Model Based Optimization

A project report submitted in partial fulfilment of the
requirements for the Award of
the Degree of

Master of Technology

in

Artificial Intelligence

By

P.SHALINI

Department of Computer & Information Sciences
School of Mathematics & Computer / Information Sciences
University of Hyderabad, Hyderabad 500046, India

June 2009

CERTIFICATE

This is to certify that the project report titled **"Model Based Optimization"** being submitted to the University of Hyderabad by ***P.SHALINI***, *Registration number 07MCMI16*, in the partial fulfillment of the requirement for the award of degree of **Master of Technology in Artificial Intelligence** is a bonafide work carried out at University Of Hyderabad under my supervision.

Prof. C.R.Rao
Project Supervisor,
Department of CIS,
University of Hyderabad

Dr. Rajeev wankar
Project Supervisor,
Department of CIS,
University of Hyderabad

Prof. Arun Agarwal
Head of Department,
Department of CIS,
University of Hyderabad

Prof. T. Amarnath
Dean,
School of MCIS,
University of Hyderabad

Acknowledgment

First and foremost, I offer my sincerest gratitude to my supervisors,**Prof.C.R.Rao and Dr.Rajeev Wankar** who has helped or supported me throughout this project with their patience and valuable suggestions. Their vast experience and profound knowledge have been a constant source for me throughout this project work. Their willingness to motivate me contributed tremendously to my project.

I am also grateful to the head of department **Prof. Arun Agarwal**, for providing excellent facilities and such a nice atmosphere for doing this project. I would like to extend my sincere thanks to Dean of school of Mathematics, Computers and Information Sciences for his valuable cooperation.

I convey my heartfelt thanks to **AI Lab staff** for allowing me to use the required equipments whenever needed.

I am extremely grateful to **D.Florance** for her constructive suggestions and cooperation throughout the project.

Finally, an honorable mention goes to my family and friends for their understanding and support in completing this project. Without help of the particular person mentioned above, I would have faced many difficulties while doing this project.

P.SHALINI

PAPER PRESENTED

P.Shalini and D.Florance, **"Image Representation: DACE Approach"**, Presented paper at **"INTERNATIONAL CONFERENCE ON OPERATION RESEACH FOR GROWING NATION"** held on 15 - 17 December 2008, Tirupati.

Abstract

The optimization of many realistic large-scale engineering systems can be computationally expensive. Standard numerical optimization techniques require large number of evaluations to provide satisfying solutions. Use of surrogate models for optimization seems beneficial. Design And Analysis Of Computer Experiments(DACE) is used to build surrogate models using kriging Approximations. DACE uses a probabilistic linear model to model a deterministic function. The maximum likelihood estimation method is used to determine parameters of this model. The predictor, which is to serve as the approximation to the original computer code is taken to be linear and unbiased. DACE was initially used for optimization process in Efficient Global Optimization (EGO).

EGO is one of the model based optimization technique developed by Schonlau(1997) and Jones et al(1998). Although EGO performs best in providing models for optimization, it would be desirable to obtain global optimum under all possible designs. Towards the end of an optimization run EGO algorithm will tend to add sample points near previous sampled points. In this report we survey about kriging and its DACE implementation and study about the existing EGO algorithm. The conventional EGO algorithm is extended by introducing the concept of symmetric points for the best point, so that the search for optimum is not restricted near previously found optimum but the entire design site is explored to get an unbiased surrogate model which converges faster to the global optimum. We study the efficiency of the EGO with symmetric points for two dimensional functions in comparison with conventional EGO algorithm in this dissertation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Optimization in mathematics is finding maximum or minimum of some function subject to some constraints. Optimization in computer science point of view is improving a system to reduce run time, bandwidth, memory requirements, or other property of a system. Optimization comprises wide varieties of techniques from operational research, artificial intelligence and computer science. Global optimization is a branch of applied mathematics and numerical analysis that deals with optimization of function or set of functions to some criteria. The most common form is the minimization of one real valued function f in the parameter-space $x^{\rightarrow} \in p$. There may be several constraints on the solution vectors space $x^{\rightarrow}_{min}$. In real-life problems, functions of many variables have a large number of local minima and maxima. Finding an arbitrary local optimum is relatively straightforward by usinglocal optimization methods. Finding the global maximum or minimum of a function is much more challenging and has been practically impossible for many problems so far. The maximization of a real-valued function g(x) can be regarded as the minimization of the transformed function f(x)=(-1)*g(x). The following section gives us a clear insight about global and local optimums

- Local Maximum: A local maximum $\hat{x}1 \in X$ of one (objective) function f: X $\rightarrow$ R is an input element with $f(\hat{x}1) \geq f(x)$ $\forall x$ neighboring $\hat{x}1$.
- Local Minimum: A local minimum $\hat{x}l \in X$ of one (objective) function f: X $\rightarrow$R is an input element with $f(\hat{x}l) \geq f(x)$ $\forall x$ neighboring $\hat{x}l$
- Global Maximum: A global maximum $\hat{x} \in X$ of one (objective) function f: X $\rightarrow$ R is an input element with $f(\hat{x}) \geq f(x)$ $\forall x \in X$.
- Global Minimum: A global minimum $x^{\vee} \in X$ of one (objective) function f: X $\rightarrow$ R is an input element with $f(x^{\vee}) \geq f(x)$ $\forall x \in X$.

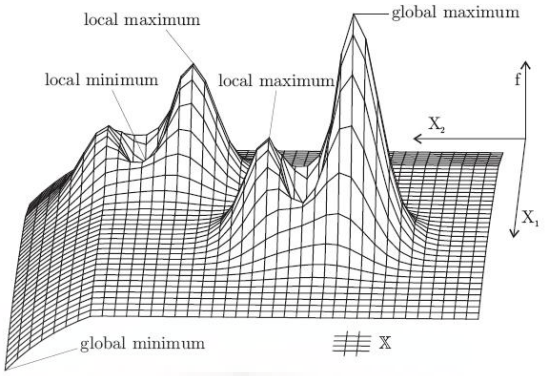


Figure 1.1: local and global optima of a two-dimensional function

1.1 Applications of Global Optimization

Typical examples of global optimization include:[1]

- Travelling salesman problem and circuit design- To minimize the path length.
- Protein structure prediction: It is the prediction of the three dimensional structure of a protein from its amino acid sequence. To minimize the free energy function
- Chemical engineering
- Worst case analysis
- Mathematical problems
- The starting point of several molecular dynamics simulations consists of an initial optimization of the energy of the system to be simulated.
- Operation research
- Networking and communication

1.2 Classification of optimization Algorithms

Figure 1.2 sketches a rough taxonomy of global optimization methods. Generally, optimization algorithms can be divided in three basic classes: according to Ref.[15]

- Deterministic: Deterministic algorithms are most often used if a clear relation between the characteristics of the possible solutions and their utility for a given problem exists. Then, the search space can efficiently be explored using for example a divide and conquer scheme. Branch and bound methods, methods based on real geometry are examples of deterministic optimization algorithms.

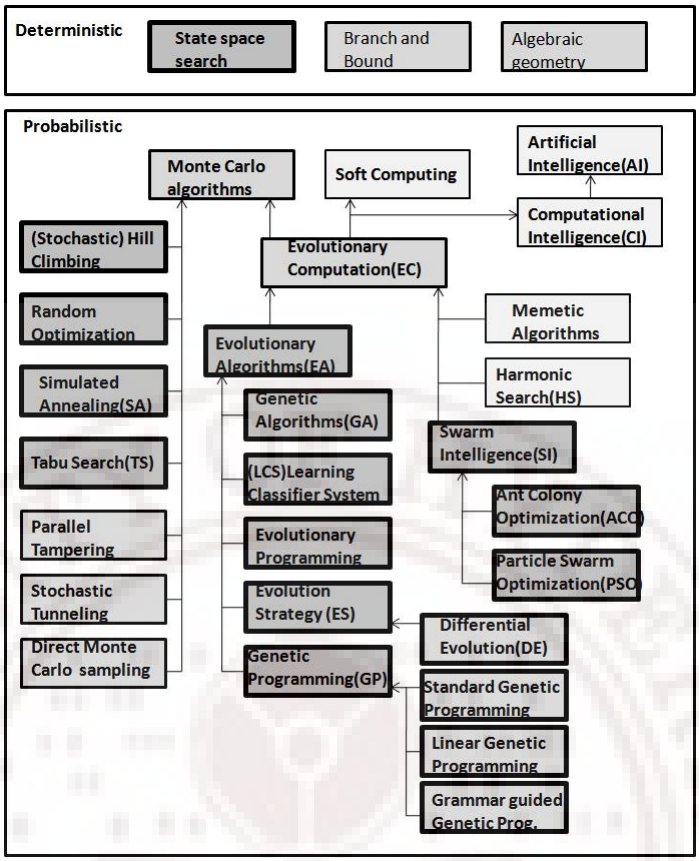


Figure 1.2: Taxonomy of global optimization algorithms

- Probabilistic or stochastic: simulated annealing, direct monte-carlo sampling comes under probabilistic optimization algorithms. They trade in guaranteed correctness of the solution for a shorter runtime. This does not mean that the results obtained using them are incorrect they may just not be the global optima.

- Heuristic and meta-heuristic: Heuristics used in global optimization are functions that help decide which one of a set of possible solutions is to be examined next. Heuristics are usually problem class dependent. A meta-heuristic is a method for solving very general classes of problems. It combines objective functions or heuristics in an abstract and hopefully efficient way, usually without utilizing deeper insight into their structure, i.e., by treating them as black-box-procedures. Examples are evolutionary algorithms, swarm based optimization algorithms

1.3 Problems in Optimization

The classification of optimization algorithms in figure 1.2 enumerate a wide variety of optimization algorithms. Why is this variety needed? One possible answer is simply

because there are so many different kinds of optimization tasks. Each of them puts different obstacles into the way of the optimizers and comes with own, characteristic difficulties.

1.3.1 Computational complexity/difficulty

The degree of difficulty of solving a certain problem with a dedicated algorithm is closely related to its computational complexity, i.e., the amount of resources such as time and memory required to do so. The computational complexity depends on the number of input elements needed for applying the algorithm. Optimization algorithms are guided by objective functions. A function is difficult from a mathematical perspective in this context if it is not continuous, not differentiable, or if it has multiple maxima and minima. In many real world applications of metaheuristic optimization, the characteristics of the objective functions are not known in advance.

1.3.2 Premature Convergence

An optimization algorithm has converged if it cannot reach new solution candidates anymore. An optimization process has prematurely converged to a local optimum if it is no longer able to explore other parts of the search space than the area currently being examined and there exists another region that contains a superior solution. There is no general approach which can prevent premature convergence. The probability that an optimization process gets caught in a local optimum depends on the characteristics of the problem to be solved and the parameter settings and features of the optimization algorithms applied. This causes loss of diversity. Losing diversity means approaching a state where all the solution candidates under investigation are similar to each other

1.3.3 Ruggedness

Optimization algorithms generally depend on some form of gradient in the objective or fitness space. The objective functions should be continuous and exhibit low total variation, so the optimizer can descend the gradient easily. If the objective functions are unsteady or fluctuating, i. e., going up and down, it becomes more complicated for the optimization process to find the right directions to proceed to. The more rugged a function gets, the harder it becomes to optimize it. If the information accumulated by an optimizer actually guides it away from the optimum, search algorithms will perform worse than a random walk or an exhaustive enumeration method.

1.3.4 Need for robustness

A system in engineering or biology is robust if it is able to function properly in the face of genetic or environmental perturbations. The goal of global optimization is to find the global optima of the objective functions. While this is fully true from a theoretical point of view, it may not succeed in practice. There will always be noise and perturbations in practical realizations of the results of optimization. There is no process in the world that is 100% accurate and the optimized parameters, designs, and plans have to tolerate a certain degree of imprecision.

1.3.5 Needle-In-A-Haystack

One of the worst cases of fitness landscapes is the needle-in-a-haystack (NIAH) problem, where the optimum occurs as isolated spike in a plane. In other words, small instances of extreme ruggedness combine with a general lack of information in the fitness landscape. Such problems are extremely hard to solve and the optimization processes often will converge prematurely or take very long to find the global optimum.

1.4 Motivation

Standard gradient-based optimization methods such as Sequential Quadratic Programming (SQP) (Schittkowski 1985) are local optimization methods and as such often have to be run many times from different starting positions to avoid local minima, and even then a globally optimum solution is not guaranteed. Approaches like Genetic Algorithm are found to be very computationally expensive. The present requirement is fast computations like online transactions which can save time and resources. Surrogate based methods are attractive because they can potentially reduce the number of calculations required and are inexpensive. An unbiased efficient surrogate model is required which overcomes the problems associated with global optimization algorithms like computational complexity and premature convergence.

1.5 Aim and Scope of the project

This project deals with finding Global optimum using cheap surrogate models and our aim is to find an unbiased surrogate model which can find global optimum in fewer number of function evaluations. Scope of this project is limited to surrogates for computer models which are used to find global optimum of the optimization functions. Computer codes are deterministic, i.e. there is no systematic, random, or human error involved in running a

computer code. The DACE surrogate model mentioned above creates an approximation from a given set of samples for such deterministic functions. DACE is used for optimization process in EGO. The conventional EGO algorithm is extended and its performance analysis is done on the different benchmark optimization test function.

1.6 Layout of the report

The layout of the report is as follows. In Chapter 2 we introduce the concept of surrogate modeling precisely, survey the various kinds of surrogate models in literature and study how surrogate model is built for deterministic computer experiments. Chapter 3 deals in depth with DACE theorm and the peculiarities of the kriging predictor. In Chapter 4 we survey different model based optimization techniques and study in detail about EGO algorithm. In Chapter 5 we discuss about different adaptive sampling criteria. Chapter 6 shows the implementation of extended EGO algorithm and chapter 7 gives the results of the implementation followed by conclusion of this report.

Chapter 2

Surrogate model

In the engineering context, experiments are always a series of tests carried out by changing system variables that are expected to have a bearing on the phenomenon being studied. Physical experiments, like wind tunnel testing for instance, require large infrastructure, careful handling and also man power. Some physical experiments, like a study of weather trends, are impossible to conduct manually. With the greater availability of computing power computer codes started being used as a convenient replacement for physical experiments. Computational Fluid Dynamics (CFD), Finite Element Method (FEM) are examples of such codes. High fidelity computer codes have now entered many areas of scientific research. With time the fidelity of computer models to nature has also steadily increased. Even with the fastest computers, computing expense is a problem in design optimization with high fidelity simulation. A need has now arisen to replace expensive computer simulations with alternative cheap surrogates in this arena. These surrogates are approximate models which replace the behavior of the original high fidelity code.

2.1 Approximation Models

A variety of approximation models and techniques exist for constructing surrogates of computationally expensive computer analysis and simulation codes. Response surface methodology and artificial neural network methods are two well-known approaches for constructing simple and fast approximations of complex computer analyses. An interpolative model known as kriging is also becoming widely used for the design and analysis of computer experiments. Multivariate adaptive regression splines and radial basis function approximations are also beginning to draw the attention of many researchers. A brief description about this approximation model is given in following section

2.1.1 Neural Networks (NN)

The idea of Neural Network (NN) came from neuroscience field which later found its application in science and engineering. With the use of initial sample points and training of NN, an approximate function can be derived which can be further used in conjunction with optimizer for optimal solution.

2.1.2 Radial Basis Function

Radial basis function was developed by Hardy (1971) and use linear combinations of a radially symmetric functions based on Euclidean or similar metric to build approximation model. A simple radial basis function form is

$$\hat{y} = \phi(x) = \sum_i \beta_i \mid\mid x - x^i \mid\mid \tag{2.1}$$

Where $|| \cdot ||$ represents the Euclidean norm, and the sum is taken over an observed set of system responses.$(x^i, f(x^i))$, i=1,..n. Replacing $\phi(x)$ with $f(x^i)$ and solving the resultant linear system yields the β_i coefficients. As commonly applied, the method is an interpolating approximation. Radial basis function approximations have produced good fits to arbitrary contours of both deterministic and stochastic response functions.

2.1.3 Response Surface Methodology (RSM)

Design of experiments (DOE) provides techniques to identify optimal parameters with a minimum of experimental effort. The most popular optimization approach is called the response surface method. The process involves the generation of functional value at multiple design points and fitting of an approximate surface through those points. In basic form least square fit is one of the examples of RSM. Theory of DOE or response surface method is based on the assumption of pure error which is modeled as independent and identically distributed (iid) normal. Response surface method constructs a smoothing fit.The curve need not pass through the exact data points. Computer experiments are deterministic. Smoothing fit for computer experiments is unacceptable.

The limitations of RSM is that the cost of fitting the approximation is proportional to the number of design variables and with low order polynomial it is quite difficult to fit a good global approximation that is valid over the entire design space. RSM is efficient only for local approximation over limited region of design space. Hence we consider interpolation methods like spline interpolation, radial basis functions and wavelets and kriging in finding optimum.

2.2 Interpolation

In the mathematical subfield of numerical analysis,interpolation is a method of constructing new data points within the range of a discrete setof known data points. In engineering and science one often has a number of data points, as obtained by sampling or experimentation, and tries to construct a function which closely fits those data points. This is calledcurve fitting or regression analysis. Interpolation is a specific case of curve fitting, in which the function must go exactly through the data points. Given a sequence of n distinct numbers x_k Called nodesand for each x_k a second number y_k, we are looking for a function fso that $f(x_k) = y_k, k = 1, ., n$ A pair x_k, y_k is called adata pointand f is called an interpolantfor the data points. For example, suppose we have a table like this, which gives some values of an unknown function *f*.

X f(x)
0 0
1 0.8415
2 0.9093
3 0.1411
4 -0.7568
5 -0.9589
6 -0.2794

Interpolation provides a means of estimating the function at intermediate points, such as x=2.5.

2.2.1 Linear Interpolation

One of the simplest methods islinearinterpolation. Consider the above example of determining $f(2.5)$. Since 2.5 is midway between 2 and 3, it is reasonable to take $f(2.5)$ midway between $f(2) = 0.9093$ and $f(3) = 0.1411$, which yields 0.5252.Generally, linear interpolation takes two data points, say (x_a, y_a) and (x_b, y_b), and the interpolant is given by:

$$y = y_a + (x - x_a)\frac{(y_b - y_a)}{(x_b - x_a)} at \tag{2.2}$$

the point (x,y). Linear interpolation is quick and easy, but it is not very precise. Another disadvantage is that the interpolant is notdifferentiable at the pointx_k.

2.2.2 Polynomial Interpolation

Polynomial interpolation replaces linear interpolant of alinear function by apolynomialof higher degree. Consider again the problem given above. The following sixth degree poly-

nomial goes through all the seven points:
$f(x) = -0.0001521x^6 - 0.003130x^5 + 0.07321x^4 - 0.3577x^3 + 0.2255x^2 + 0.9038x$.
Substitutingx= 2.5, we find that$f(2.5) = 0.5965$. Generally, if we havendata points, there is exactly one polynomial of degree atmostn-1 going through all the data points. The interpolation error is proportional to the distance between the data points to the powern. Furthermore, the interpolant is a polynomial and thus infinitely differentiable. So, we see that polynomial interpolation solves all the problems of linear interpolation. However, polynomial interpolation also has some disadvantages. Calculating the interpolating polynomial is computationally expensive compared to linear interpolation. Furthermore, polynomial interpolation may not be so exact especially at the end points.

2.2.3 Spline Interpolation

Spline interpolation uses low-degree polynomials in each of the intervals, and chooses the polynomial pieces such that they fit smoothly together. The resulting function is called a spline. For instance, the natural cubic splineispiecewisecubic and twice continuously differentiable. Furthermore, its second derivative is zero at the end points. The natural cubic spline interpolating the points in the table above is given by

$$f(x) = \begin{cases} -0.1522x^3 + 0.9937x, & if x \in [0,1], \\ -0.01258x^30.4189x^2 + 1.4126x - 0.1396, & if x \in [1,2], \\ 0.1403x^3 - 1.3359x^2 + 3.2467x - 1.3623, & if x \in [2,3], \\ 0.1579x^3 - 1.4945x^2 + 3.7225x - 1.8381, & if x \in [3,4], \\ 0.05375x^3 - 0.2450x^2 - 1.2756x + 4.8259, & if x \in [4,5], \\ -0.1871x^3 + 3.3673x^2 - 19.3370x + 34.282, & if x \in [5,6] \end{cases} \tag{2.3}$$

In this case we getf(2.5)=0.5972.

2.2.4 Interpolation VIA Gaussian processes

Gaussian processis a powerful non-linear interpolation tool. Many popular interpolation tools are actually equivalent to particular Gaussian processes. Gaussian processes can be used not only for fitting an interpolant that passes exactly through the given data points but also for regression, i.e. for fitting a curve through noisy data. In the geostatistics community Gaussian process regression is also known as Kriging

2.3 Surrogate modeling philosophy

Computer codes are black boxes i.e. usually no analytical expression is available for their output. Hence no a priori knowledge of the output variation with change in input is avail-

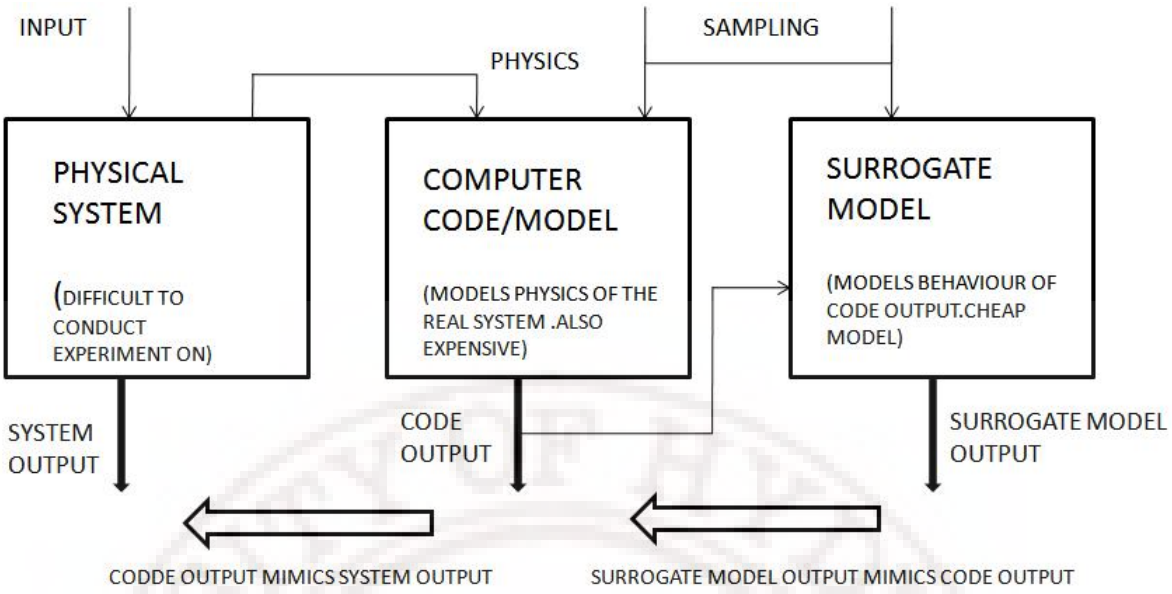


Figure 2.1: surrogate modeling philosophy

able. Computer codes are deterministic, i.e. there is no systematic, random, or human error involved in running a computer code. The Design and Analysis of Computer Experiment (DACE) surrogate model creates an approximation from a given set of samples for such deterministic black box functions. As shown in Figure (2.1) The original computer code has the information of the physics of the system, but the surrogate model gets information about the output of the code only by sampling it.

The objective of the surrogate model as shown in fig(2.1) is to predict the values of deterministic function y(x),$x \in R^n$,$y \in R^q$, over a variable space D, $D \subset R^n$ when its values are known only at a limited, finite number of sites contained in S=$\{s_1, s_2, \cdots, s_i \in D\}$.To make a good surrogate model we can choose S in many different ways. m is usually bounded by operational constraints. Finding these design sites and a suitable surrogate model is the objective of DACE. According to [5] the problem of creation of a surrogate model for a computer experiment can be subdivided into 2 parts:

1. The Design problem: At which sites in S=$\{s_1, s_2, \cdots, s_i \in D\}$ should the output data Y=$[y(s_1), y(s_2), \cdots, y(s_m)]^T$ be collected?

2. The Analysis problem: How should the data be used to make a surrogate model that will help us predict y(x) $\forall x \in D$with reasonable accuracy?

This methodology is applicable only when y(x) is a deterministic function.
Surrogate model built for computer experiments is fair and reusable. The surrogate model gives us the parameters of the fitted function. This information provided by the model can be used for various purposes. Surrogate model can be used for representations. Representation of image using DACE model is a novel approach in image formats. This work

is presented in an INTERNATIONAL CONFERENCE ON OPERATION RESEARCH FOR GROWING NATION held on 15-17 December 2008 and paper is entitled as "IMAGE REPRESENTATION : DACE APPROACH" by me and D.Florance. The detail work on image representation through DACE model is given in dissertation written by D.Florance. The other major area where surrogate model is used is in field of optimization which is demonstrated in this dissertation. The philosophy of DACE and building kriging model is the subject matter of coming chapter.

Chapter 3

DACE

In structural optimization, usually an approximation concept is introduced as interface between structural analysis code and optimization algorithm. In some cases of optimum design, a global approximation concept can be effectively applied. Then, approximation model functions are built of all objective and constraint functions, whose values, for a certain design point, follow from the structural analysis calculations. In this way, the original optimization problem is completely replaced by an explicitly known approximate optimization problem.
Sacks and coworkers proposed a new model building strategy, which is especially suited for deterministic computer responses [5]. Their basic assumption is that computer responses can be modeled as a realization of a stochastic process. This finally leads to a response prediction that exactly describes all calculated computer responses. Additionally, the mean squared error of a prediction can be calculated, which serves as a measure of accuracy of the prediction. Main advantage of the approximations of Sacks et al., compared with response-surface models, is the flexibility to automatically adapt to the calculated response data.

3.1 Kriging

Kriging is basis of both regression models and radial basis functions. In this approach the underlying process is assumed to be a superposition of a linear model and departures from the linear model
Actual Process = Linear model + Systematic departures
Such a model is also called a "probabilistic linear model". Based on the knowledge of the underlying process being studied, the systematic departures are modeled as stochastic process of a certain kind. The approach of using a realization of a stochastic process has

traditionally been used in geostatistics under the name of Kriging. Derived from a miner named Krige, Kriging now is synonymous with spatial prediction. The linear model is a taken as regression model. Thus,

$$Linearmodel = \sum_{i=1}^{p} f_i \beta_i \tag{3.1}$$

Where β_i's are regression parameters and f_i's are regression functions. β is found as least square solution of regression problem. The stochastic process is taken to be a decaying function of the distance from a point. Hence Kriging also qualifies as a type of radial basis function surrogate model. And it indeed produces an approximation of the kind

$$y(x) \approx f(x)^T \beta + \Gamma(x)^T \gamma \tag{3.2}$$

Depending on the choice of the stochastic process we get different kinds of Kriging approximations. There are several types of Kriging methods- simple Kriging, ordinary Kriging, indicator Kriging, universal Kriging to name a few. The entire framework of universal Kriging was put into a form that could be used to make approximations to expensive and complex computer models. In some sense this framework provides a "model of a model"- a function surrogate model of an expensive physical surrogate model. Such a model is often referred to in literature as a metamodel. This framework is also called Design and Analysis of Computer Experiments (DACE), after the paper in [5] by Sacks et al, . The reason why DACE is amenable to computer models, not to physical systems is that it is applicable only to deterministic systems. Essentially DACE can be applied to all kinds of processes that are deterministic.

3.1.1 Types of kriging

Depending on the stochastic properties of the random field different types of kriging apply. The type of kriging determines the linear constraint on the weights w_i implied by the unbiasedness condition; i.e. the linear constraint, and hence the method for calculating the weights, depends upon the type of kriging. Classical types of kriging are:

- Simple kriging: Simple krigingassumes a known constant trend: $\mu(\mathrm{x}) = 0$. Simple kriging is mathematically the simplest, but the least general. It assumes the expectation of the random fieldto be known, and relies on a covariance function. However, in most applications neither the expectation nor the covariances are known beforehand.

- Ordinary kriging: Ordinary kriging is the most commonly used type of kriging. It assumes a constant but unknown mean. $\mu(\mathrm{x}) = \mu$.

- Universal kriging: Universal kriging assumes a general linear trend model. $\mu(x) = \sum_{k=0}^{p} \beta_k f(x)$
- Indicator kriging: Indicator kriging uses indicator functions instead of process itself, in order to estimate transition probabilities.

Modeling and prediction of computer data using universal kriging and DACE is discussed in the next section.

3.2 Theorm: Modeling and Estimation

Modelling of the deterministic computer response y(x) is a realization of stochastic process Y:

$$y(x) = \sum_{i=1}^{k} f_i(x)\beta_i + z(x) \tag{3.3}$$

Where f_i's are known regression function,β_i's are unknown regression parameters.
The model is sum of linear regression model $f^T(x)\beta$ and a random process Z(x).This random process is assumed to have zero mean and covariance between Z(x) and Z(w) At design sites x and w is product of process variance σ^2 and a correlation function R(w, x)

$$cov(w, x) = \sigma^2 R(w, x) \tag{3.4}$$

The stochastic process Y is assumed to be Gaussian.
For a certain experimental design $S = \{s_1, \cdots, s_N\}$, computer experiments have been performed, and response data $y_s = \{y(s_1), \cdots, y(s_N)\}$ is available. From these computer responses unknown parametes β and σ^2 are estimated. The correlation function R(w, x) is given by

$$R(w, x) = \prod_{j=1}^{n} R_j(d_j) \tag{3.5}$$

With: $d_j = w_j - x_j$

$$R_j(d_j) = e^{-\theta_j |d_j|^{p_j}} \tag{3.6}$$

In compact Eq.(3.3) can be written as

$$y(x) = f(x)^T \beta + z(x) \tag{3.7}$$

$f(x) = [f_1(x), f_2(x), \cdots, f_k(x)]^T$,$\beta = [\beta_1, \beta_2, \cdots, \beta_k]^T$, $\theta = [\theta_1, \theta_2, \cdots, \theta_n]^T$
Z(x) is function of σ^2 and θ. The task now is to find the parameters β, σ^2 and θ which is posed as maximum likelihood. The maximum likelihood estimation technique and its use for kriging is described in appendix 1

3.3 Kriging Predictor

The predictor of universal kriging according to [11],

1. Linear with respect to the output data
2. Unbiased.

Linearity implies that the predictor must be of the form given below.

$$\hat{y}(x) = c(x)^T Y \tag{3.8}$$

Unbiasedness implies that

$$E(\hat{y}(x)) = E(y(x)) \tag{3.9}$$

For any given data, there exist infinite predictors that are both linear and unbiased. The Best Linear Unbiased Predictor (BLUP) is the one that gives minimum mean square error between the predictor and the function. Thus finding BLUP requires that we

1. Minimize $E(\mid \hat{y}(x) - y(x) \mid^2)$ with respect to c(x)
2. subject to $E(\hat{y}(x)) = E(y(x))$

The set of conditions above now form constrained optimization problem, which is solved by using langrange multipliers in Appendix 2.Solution of the problem will yield us BLUP for data S as

$$\hat{y}(x) = \Gamma(x)^T R^{-1} Y - (F^T R^{-1} \Gamma(x) - f(x))^T (F^T R^{-1} F)^{-1} F^T R^{-1} Y \tag{3.10}$$

Finally kriging predictor is given by

$$\hat{y} = f^T(x)\hat{\beta} + \Gamma^T(x)\hat{\alpha} \tag{3.11}$$

Where

$$\hat{\alpha} = R^{-1}(y_s - F\hat{\beta}) \tag{3.12}$$

With Γ being column of correlations between the Z's at design sites S and untried inputs X:

$$\Gamma = [R(s_1, x), \cdots, R(s_N, x)]^T \tag{3.13}$$

The second part of the Eq.(3.7) is an interpolation of the residuals of the regression model $f^T(x)\beta$. Therefore, all the response data will be exactly predicted.$\hat{\beta}$ and $\hat{\alpha}$ are independent of untried points x. Hence as seen in Eq. (3.11) evaluation of $\hat{y}(x)$ for every new x involves only the computation of $f(x)^T$ and $\Gamma(x)^T$ which are much easier to compute than the

original function. This is the huge advantage of creating a computer model.
Sacks et al.(1989b) also gave representation of mean squared error:

$$MSE(\hat{y}(x)) = \sigma^2 \left\{ 1 - [f^T(x)\ \Gamma^T(x)] \begin{bmatrix} 0 & F^T \\ F & R^{-1} \end{bmatrix} \begin{bmatrix} f^T(x) \\ \Gamma^T(x) \end{bmatrix} \right\} \tag{3.14}$$

MSE is a measure of uncertainty associated with the predicted value. It is remarked that Mean square error is zero for X equal to design sites in S. A greater value of MSE at a point implies that the underlying process is inadequately represented by the samples in the region around that point.

3.4 MSE

Mean square error is regarded to be confidence or the uncertainty that one can place in the values predicted by kriging predictor $\hat{y}(x)$. For a black box function whose analytical form is not known, the MSE is calculated using probabilistic linear model for y(x) in Eq.(3.7) as

$$MSE(x) = E(\mid \hat{y}(x) - y(x) \mid^2) \tag{3.15}$$

The parameters of this model are calculated using existing samples. Hence MSE found is not the real error between the predictor and actual underlying process. It is simply error between predictor and model in Eq.(3.7).How closely MSE resembles the real error depends on how well we have sampled the existing data. A low value of MSE does not necessarily imply that real error is small. Ref.[12] provides some discussion on aliasing in surrogate models.

Chapter 4

Model Based Optimizaion

In many engineering optimization problems, the number of function evaluations is severely limited by time or cost. These problems pose a special challenge to the field of global optimization. In mechanical engineering and particularly in the field of machining, improvements in materials, coatings, tools, and machines constantly offer increased productivity. In order to completely exploit these potentials, an optimal set-up of the resulting processes has to be found. Since standard numerical optimization techniques such as evolutionary algorithms, swarm optimization, and numerical approaches require a large amount of evaluations to provide satisfying solutions, and experiments are time-consuming and expensive, the use of a surrogate model for the real process seems beneficial. According to Ref.[2] If enough knowledge about the process and the involved material exists to create an appropriate physical model, simulation tools can be used to create a surrogate of the process. Otherwise, if such knowledge does not exist and the design of an independent simulation is too complex, empirical modeling is the only possibility to summarize the information obtained from experiments, and efficiently optimize the underlying process.

4.1 Modeling and Simulation

While there are a lot of problems where the objective functions are mathematical expressions that can directly be computed, there exist problem classes far away from such simple function optimization that require complex models and simulations.

Model: A model is an abstraction or approximation of a system that allows us to reason and to deduce properties of the system. Models are often simplifications or idealization of real-world issues [15]. They are defined by leaving away facts that probably have only minor impact on the conclusions drawn from them. In the area of global optimization, we often need two types of abstractions:

1. The models of the potential solutions shape the problem space X. Example programs

in Genetic Programming, like the Artificial Ant problem.

2. Models of the environment in which we can test and explore the properties of the potential solutions, like a map on which the Artificial Ant will move which is driven by the evolved program

Simulation : A simulation is the computational realization of a model. Whereas a model describes abstract connections between the properties of a system, a simulation realizes these connections. Simulations are executable, live representations of models that can be as meaningful as real experiments. They allow us to reason if a model makes sense or not and how certain objects behave in the context of a model.
The optimization based on a model of an empirical process is quite different from a simulation-based optimization [2]. While the simulation is evaluated online by the optimization algorithm potentially assisted by local meta-models to avoid unnecessary evaluations only a few stages of real experiments can be carried out. If the number of different designs is known before experimentation, a lot of time can be saved since the necessary resources can be estimated and the order of experiments can be appropriately planned. Therefore, the main experimental effort is spent on the generation of the initial model. After this, only a few further experiments can be conducted to refine the model and to validate the identified optimum.
Different model based optimization techniques are Efficient Global Optimization (EGO), Sequential Parameter Optimization (SPO) and Fixed Nugget Optimization (FINO). The following section gives description about various model based optimization techniques.

4.2 Sequential Parameter Optimization

SPO is a heuristic that combines classical and modern statistical techniques to improve the performance of search algorithms.SPO consists of 3 iterative stages. The first stage determines design points. Latin hypercube sampling and model based selection is used to choose new points. The best point found so far is reevaluated. During the second stage designs points are evaluated by running the algorithm using the proposed setting. During the third stage a model is build, to estimate the algorithms performance for untested settings. Results of second and third stage may be of interest to the user. As experimental results are obviously interesting for the user, also the model could show insightful properties of parameters and their relation.
Sequential Parameter Optimization (SPO) tries to upgrade EGO to parameter optimization of stochastic search algorithms, where the evaluations are highly distorted by random effects. It uses a quadratic regression model, fixes Polynomial degree to two, and performs

multiple reevaluations of each parameter setting to avoid bad predictions due to noise-induced false estimations of the correlation models parameters.
However, there are some drawbacks in SPO. SPO is constrained to decimal and integer values which is an obvious limitation. If number of combinations of nominal or ordinal parameters is low, optimization can be done for each combination and choose the best one. Another difficulty when applying SPO stems from the fact that it also requires the specification of some parameter values. Concerning this issue, there are currently no theoretical results available, so that one has to rely on experience from previous experimental studies.

4.3 Fixed Nugget Optimization

A DACE variant capable of directly dealing with noise in the measured data is provided in the kriging literature of geostatistics by means of the so-called nugget. In case of semivariogram which is non continuous only at the origin, the height of the jump at the origin is referred to asnuggetor nugget effect. A new variant of the modeling approach used in EGO, in which a nugget factor is applied to the general correlation model is called Fixed Nugget Optimization (FINO).
Ref.[2] says that in Fixed Nugget Optimization (FINO), the standard deviation of the measured observations $\hat{\sigma}$ is estimated by repeating the experiment of the initial design, which obtains the median result in terms of the objective, three times. Then, $\hat{\sigma}$ is used to compute the nugget using the formula $c_{nugget} = \frac{\hat{\sigma}^2}{\sigma^2+\hat{\sigma}^2}$, where σ^2 is the process variance. This approach is closely related to the central composite designs used in DOE, where only the center point is repeated to obtain an estimate of the process-induced noise. Thus, even for noisy function evaluations, the DACE variant proposed in EGO, which is designed for deterministic computer experiments, provides significantly better models in cases where the functional relation, which is fitted for the DOE model, is not exactly appropriate.

4.4 Efficient Global Optimization

In the last decades alternative methods to generate empirical surrogate models have emerged. One of the most popular method among them is DACE (Design and Analysis of Computer Experiments). The DACE concept was first used for optimization purposes in EGO (Efficient Global Optimization). EGO is one of the model based optimization technique. The Efficient Global Optimization (EGO) technique was tossed by Schonlau 1997; Jones et al.1998. It is preferred on other model based optimization techniques because of its ease of implementation and conceptual simplicity.

4.4.1 The EGO Algorithm

The EGO algorithm available in literature proceeds as follows. A random sampling of the input parameters is generated. Here the 25 random samples are generated by Improved Hypercube Sampling (IHS), which attempts to generate a space filling design, but any suitable design of experiments method could be used. The objective function is then evaluated for each input and a surrogate model is fitted.

This surrogate model describes both the variation of the mean value of the objective function between the sample points and the uncertainty between them, and is much less computationally expensive to evaluate than the original objective function. In this application, a Kriging technique is used. Another appropriate optimization technique such as, simulated annealing or the DIRECT method is then employed to find the next best place to sample for a minimum objective function. The secondary objective function used in this application is the Expected Improvement (E[I]) objective function. The improvement function (I) is defined as the improvement of the current prediction, $\hat{y}(x)$, at point x over the minimum value of the current set of samples, y_{min}, i.e.

$$I = max(y_{min} - \hat{y}(x), 0) \tag{4.1}$$

The expected improvement, defined as the expectation of the improvement, is given by

$$E[I] = (y_{min} - \hat{y}(x))CDF\left(\frac{y_{min} - \hat{y}(x)}{s(x)}\right) + s(x)PDF\left(\frac{y_{min} - \hat{y}(x)}{s(x)}\right). \tag{4.2}$$

Where CDF(x) is the standard normal cumulative density function, and PDF(x) is the standard normal probability density function. The Expected Improvement represents the potential of a new observation of the objective function to be smaller than the existing minimum observation. The point at which the value of the Expected Improvement is maximized gives the best point at which to calculate the true objective function. By considering the possibility that areas having a large predicted mean but also a large predicted variance (the second term in Eq(4.2)) might be preferable to search than areas that have a small predicted mean but also a small predicted variance (the first term in Eq(4.2)), the function searches for both local and global minima. The surrogate model is then updated to include the newest sampled point, and the operation repeated until the sampling point does not change and the global minimum of the objective function has been found. The algorithm according to Ref.[8] appears below

1. An initial set of input parameters is selected using IHS.
2. The true objective function is evaluated for all new members of the set.
3. A Kriging surrogate model is fit to the values of the objective function.

4. The expected improvement objective function, calculated using values from the computationally inexpensive Kriging model, is minimized using any suitable global optimization method.

5. The result of the minimization (the next input parameters most likely to improve the true objective function) is added to the set.

6. The process repeats from step 2 until a predetermined number of iterations is reached.

One advantage of the EGO method is that it requires a minimal number of objective function evaluations, and most of the optimization is done on the computationally cheap surrogate. This makes the method very efficient when the objective function is computationally expensive. There is no proof of global convergence for the EGO algorithm, but a discussion and conjecture on this matter is given in Jones et al. (1998). The minimization of expected improvement objective function is chosen as the point to be added in the next iteration in EGO algorithm. Different criteria for adding the sample points in successive iterations are discussed in the next chapter.

Chapter 5

Adaptive Sampling Strategy

The general principle of optimization using Kriging is iteratively to evaluate f at a point that optimizes a criterion based on the model obtained using previous evaluation results. The simplest approach would be to choose a minimizer of the prediction f as a new evaluation point. However, by doing so, too much confidence would be put in the current prediction and search is likely to stall on a local optimum. To compromise between local and global search, more emphasis has to be put on the prediction error that can indicate locations where additional evaluations are needed to improve confidence in the model. This approach has led to a number of criteria, based on both prediction and prediction error, designed to select additional evaluation points.
In (Jones [2001]) and (Watson and Barnes [1995]), a fair number of alternative criteria are presented and compared. Although quite different in their formulation, they generally aim to answer the same question: What is the most likely position of x^* ? Another, and probably more relevant, question is: Where should the evaluation is carried out optimally to improve our knowledge on the global minimizers?
In the case of a black box computer model, where the function topology is not known, no proactive sampling strategy can be relied upon for a good approximation. Hence none of the conventional space filling strategies like random sampling, uniform sampling, Latin hypercube sampling can be used effectively with all the m points. The only information we have about the actual function is from the existing samples and the DACE predictor created out of them. Hence it is essential to use a reactive strategy which learns from the information provided by previous samples to get the new set of samples. Such a process is commonly referred to as adaptive sampling. Adaptive sampling as a technique often appears in the area of clinical research.
A general framework for DACE with adaptive sampling is as follows:

1. Use some space filling sampling strategy to sample some points in the variable-space

2. Use DACE to fit a surrogate model
3. Find the new sample by satisfying some sampling criteria.
4. Repeat step 2 & 3 with new set of samples
5. Stop when some termination criterion is reached.

5.1 Issues arising in adaptive sampling

Some issues emerge out of the inspection of this framework.

1. What should be the initial sampling strategy?
2. What strategy is used in adding the sample point in the next iteration?
3. What is the measure of accuracy of a model?
4. What is the guarantee of convergence of the algorithm vis--vis the initial sampling strategy and additional of sample points in the next iterations?

5.2 Initial Sampling strategies

The decision of the initial sampling strategy can be decomposed into two choices

(a) The type of sampling, i.e. the distribution of points in the design space.

(b) Number of points (out of m) to be sampled with respect to the maximum number of samples.

Both (a) and (b) are heavily dependent on the users knowledge of the underlying process in the computer model. In the case where no knowledge is available about the underlying process, the user is best placed to use any of the standard space filling strategies. Some popular space filling sampling strategies are

- Random sampling- randomly chooses points over the entire domain.
- Uniform sampling- fills the space with uniformly spaced points
- Latin hypercube sampling chooses points such that no two of them have the same coordinate, and fills the space thus.
- Grid sampling uses a systematic approach that divides the field into squares or rectangles of equal size (usually referred to as grid cells).

5.3 Criteria for adding next sample point

A variety of different criteria have been used in literature. We shall survey them here in brief. Ref.[6] uses adaptive sampling to find the global optimum of the black box function in EGO. It uses a function called expected improvement which is a probabilistic measure of the chances of that the new global minimum is lower than the current minimum sampled function value y_{min}.

$$E[I] = (y_{min} - \hat{y}(x))CDF\left(\frac{y_{min} - \hat{y}(x)}{s(x)}\right) + s(x)PDF\left(\frac{y_{min} - \hat{y}(x)}{s(x)}\right). \tag{5.1}$$

Where CDF(x) is the standard normal cumulative density function and PDF(x) is the standard normal probability density function. The next sample point is added at global maximum of E[I]. When looking for the next sample there is always a dilemma about choosing between:

(a) Searching in the neighborhood of y_{min} for a better minimum by sampling in the vicinity of y_{min} and

(b) Searching in region that is sparsely sampled to check for possibilities of a better global minimum in that region.

Disregarding either (a) or (b) would be using the known information insufficiently. It is necessary to balance both (a) and (b) which is what E[I] does. In the same vein as EGO, superEGO uses a generalized expected improvement [9]. By introducing a non-negative integer parameter g the generalized improvement is defined as

$$I_g(x) = max\left\{0, (f_{min} - y(x))^g\right\} \tag{5.2}$$

The value of g determines which of the trends (a) and (b) is given more preference. Ref.[10] has used the adaptive sampling in the EGO framework with Kriging approximation to improve the design of ergonomics experiments. The criterion used by [10] is to maximize the MSE of the DACE model. However this criterion is applied selectively only in certain regions of the design space that are most relevant to their experiment. This is to say the new sample x_{new} is given by

$$x_{new} = argmax_{x \in D}(MSE(x)) \tag{5.3}$$

It is crucial that the iterative sampling criteria are well catered to the kind of problem we are trying to solve. Existence of better sampling criteria for the next iteration is always matter for further research. According to [9] there are no rigorous convergence criteria for the adaptive sampling technique used by EGO and superEGO. Extension of this adaptive sampling strategy is shown in the next chapter.

Chapter 6

Extended EGO algorithm

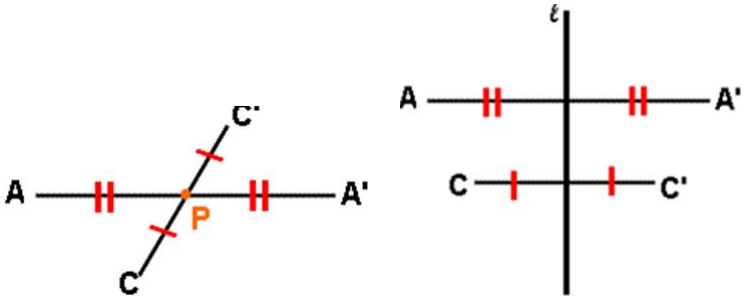
In the chapter of adaptive sampling we have seen different criteria for adding sampling points. It is noticed that the expected improvement criteria of EGO algorithm balance well between both the conditions of adding sample points for the successive runs. Although EGO performs best in providing models for optimization, it would be desirable to obtain stable results under all possible designs [2]. EGO algorithm sometimes takes more number of evaluations to get global optimum. In some cases it even fails to converge to the global optimum. Ref. [8] says that there is no proof of global convergence for the EGO algorithm. This is because towards the end of an optimization run, the conventional EGO algorithm will tend to sample points near previously sampled points [6].Iterative evaluation of DACE model by adding sample points in search of optimum tends to lose its symmetric nature. To nullify biased nature of conventional EGO algorithm the appropriate modifications in adaptive sampling strategies is essential.

6.1 Concept Of Symmetric Points

By analyzing biased approach of conventional EGO it is proposed to have adaptive sampling by introducing symmetricity.

SYMMETRIC: An object is symmetric with respect to a given operation if this operation, when applied to the object, does not appear to change it. A mathematical operation, or transformation, that results in the same figure as the original figure or its mirror image is called a symmetry operation.Such operations include reflection, rotation, double reflection, and translation. in general, a symmetry operation on a figure is defined with respect to a given point (center of symmetry), line (axis of symmetry), or plane (plane of symmetry).

LINE SYMMETRY: A figure in the plane has a line of symmetry if the figure can be mapped onto itself by a reflectionin the line. Line symmetry,or justsymmetry, occurs when two halves of figure mirror each other across a line. Theline of symmetry is the line that



(a) This is point symmetry (b) This is line symmetry

Figure 6.1: Figure Of POINT and LINE symmetry

divides the figure into two mirror images. Line symmetry is shown in Figure(6.1(b))

PLANE SYMMETRY: Plane symmetry occurs when a plane intersects a three dimensional figure such that one half is the reflected image of the other half. Such a plane is imaginary and divides an object into two halves, each of which is the mirror image of the other in this plane. Plane symmetry is analogous to line symmetry, only in three dimensions. Common objects displaying plane symmetry are rectangular solids, spheres, boxes, cones, and humans.

POINT SYMMETRY: Point symmetry exists when a figure is built around a single point called the center of the figure. For every point in the figure, there is another point found directly opposite it on the other side of the center. An object has rotational symmetryif there is a center point around which the object is turned (rotated) a certain number of degrees and the object looks the same. The number of positions in which the object looks exactly the same is called the orderof the symmetry. Figure (6.1(a)) shows point symmetry.

6.2 Extended EGO algorithm with symmetric points

In our extended EGO algorithm we proceed with the objective of finding global maximum. For a fixed domain initially few sample points are taken. For those sample points the function is evaluated and their responses are calculated. For these sample points DACE model is fit with the help of some regression and correlation parameters and using kriging approximations. Figure(6.2) shows block diagram of implementation. With the help of this model for the additional points in the grid, the model is able to predict the response and also the Mean Square Error (MSE) associated with the prediction. Maximum of initial sample points is calculated. Then the additional sample point for the next iteration is added by calculating either using probability or MSE as shown in Equations (6.1) and

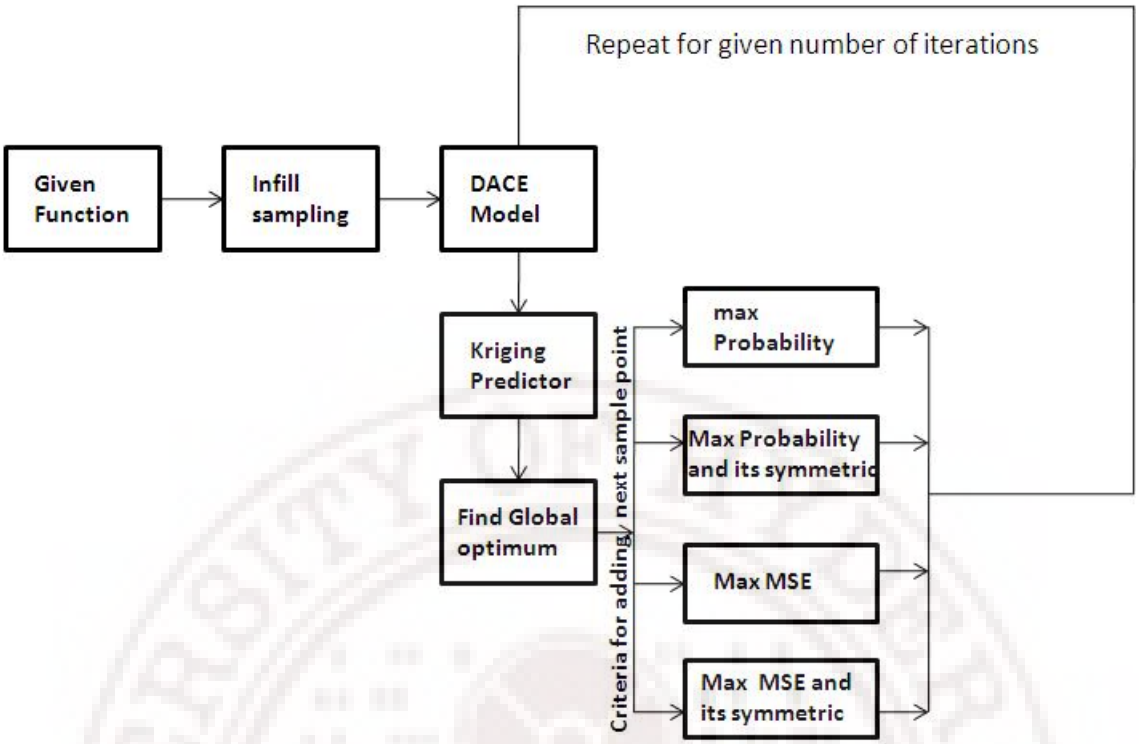


Figure 6.2: Block Diagram of Implementation

(6.2) respectively

$$Probability = 1 - CDF(y_{max}, \hat{y}) \tag{6.1}$$

This probability as in the case of conventional EGO finds out the chance that there exists a point whose maximum is more than the existing maximum.

$$x_{new} = argmax_{x \in D}(MSE(x)) \tag{6.2}$$

MSE is the error associated with the predicted value. This is the point where MSE is maximum as studied in the previous chapter.

In the extended EGO algorithm we add symmetric points for the best point found using probability or MSE. For a two dimensional function 4 symmetric points can be added including that point i.e., symmetry about x-axis, symmetry about y-axis and symmetry about origin. Similarly for a three dimensional function 8 symmetric points can be added including that point i.e., symmetry about x-axis, y-axis, z-axis, xy-plane, yz-plane, zx-plane and symmetry about origin. So, in general for a n-dimensional function 2^n symmetric points can be added including the best point found.

6.2.1 Convergence

As we have seen in the introduction chapter of optimization premature convergence is one of the problems faced by the numerical optimization approaches. According to [9] there

is no rigorous convergence criteria for the adaptive sampling technique used by EGO. [11] suggests that sampling may be stopped once the improvement of the current best sample becomes sufficiently small. In our case, where we add symmetric points for the best point found, the most prudent termination criteria is the chance of finding the best point becomes less than sufficiently small, less than a threshold value.

6.2.2 Performance Analysis

Few benchmark test functions of optimization are taken. The efficiency of the extended EGO algorithm with symmetric points is tested on the conventional EGO algorithm with the help of these test functions. Efficiency is tested on two bases: number of iterations taken by the function to converge to global maximum and the number of points at which function is evaluated in the process of reaching global maximum. For some functions which has failed to converge to global optimum we have calculated the error rate by which it lacks in converging to global maximum. Error rate is calculated as

$$\text{Error rate} = \frac{(\text{Global optimum- converged optimum})}{\text{Global optimum}} \tag{6.3}$$

.

Chapter 7

Results

The results of our extended EGO algorithm are shown in this chapter. We took fifteen unconstrained benchmark test functions of optimization and analyzed the performance of our extended EGO algorithm over conventional EGO algorithm. These test functions are from three different classes and all of them are continuous.

(a) unimodal and convex.

(b) multimodal, two-dimensional with a small number of local extremes.

(c) multimodal, two-dimensional with large number of local extremes.

Each test function is run for different cases for comparing the results and efficiency of our extended EGO algorithm. We have tested every optimization test function for four different cases:

1. Conventional EGO which considers the maximum probability as the best point to be added in the next iteration.

2. Extended EGO algorithm which adds four symmetric points for one best point found using maximum probability.

3. Considering the point where MSE is maximum as the next sample point to be added.

4. Adding four symmetric point for the best point found using maximum MSE.

When applied our extended EGO algorithm on these fifteen benchmark functions some of the functions have shown equal performance with the conventional EGO algorithm. The performance is indistinguishable because global optimum for those functions lie at the point where initial uniform sampling is done. Eight functions out of fifteen functions showed significant change in the performance when our extended EGO algorithm is applied over conventional EGO algorithm. First we demonstrate the results of a one-dimensional

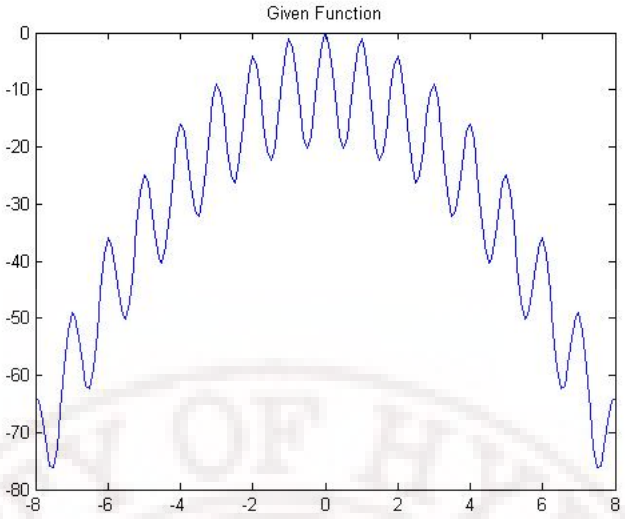


Figure 7.1: Rastringin function in one-diimensional

function in the next section followed by two-dimensional functions. The results of the five benchmark test functions are shown under two-dimensional functions.

7.1 One-dimensional function

In case of one dimensional functions extended EGO algorithm adds $2^1 = 2$ points at the end of each iteration i.e., symmetric about origin whereas conventional EGO algorithm adds 1 point. To demonstrate the working and results of the EGO and Extended EGO algorithm in the case of one-dimensional functions lets consider the Rastringin‘s function.

7.1.1 Rastringin‘s function

Rastrigin‘s function is based on the function of De Jong with the addition of cosine modulation in order to produce frequent local minima.The test function is highly multimodal. However, the location of the minima are regularly distributed. Function has the following definition

$$f(x) = 10n + \sum_{i=1}^{n}[x_i^2 - 10cos(2\pi x_i)] \tag{7.1}$$

The test area for the function is restricted to $-8 \leq x \leq 8$. Its global maximum is f(x)=0 obtainable at x_i=0.Rastringin‘s function in one dimension is as shown in fig(7.1). Uniform sampling is done within the test area with different sample rates of 2,4 and 8. For example, when initial sample rate is taken as four five points are selected initially from -8 to 8.Those points are -8,-4,0,4,8. The initial sample points and the predicted function with the help those sample points are as shown in fig(7.2).

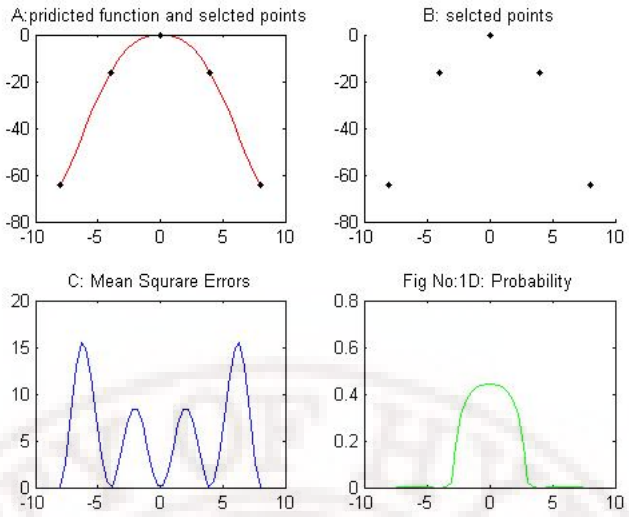
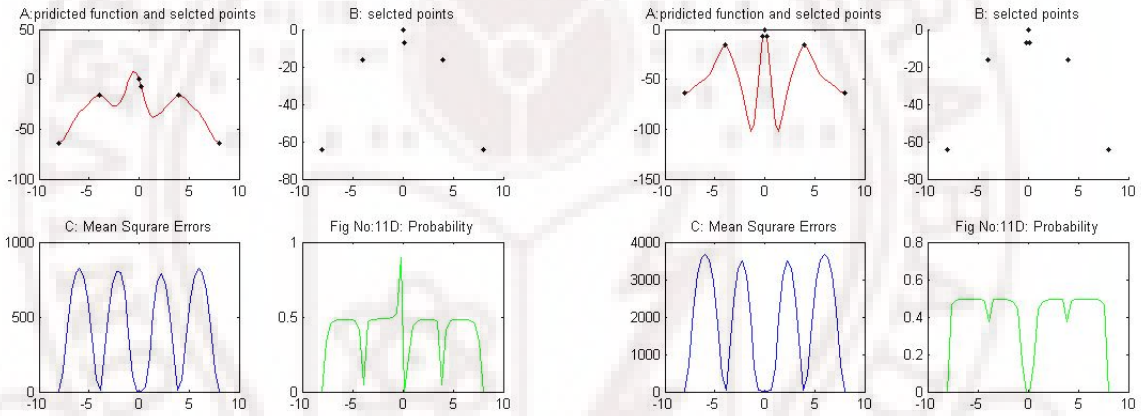


Figure 7.2: Initial sample points and predicted function of Rastringin function



(a) EGO 1st iteration

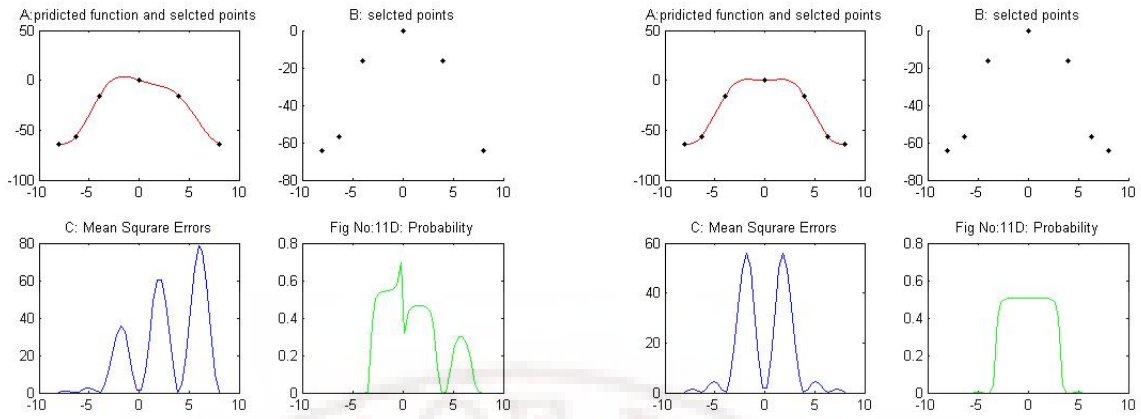
(b) extended EGO 1st iteration

Figure 7.3: Comparing EGO and Extended EGO for 1st iteration for Rastringin function

With the help of these initial sample points DACE model is built. A Grid of forty points are generated by grid sampling within the test area of -8 to 8.Response of the rastringin test function is predicted at those forty points with the help of the DACE model built using the initial sample points. Now let us see in detail how the next sample points are added in successive iterations while using EGO, extended EGO, MSE and MSE with symmetric points algorithms.

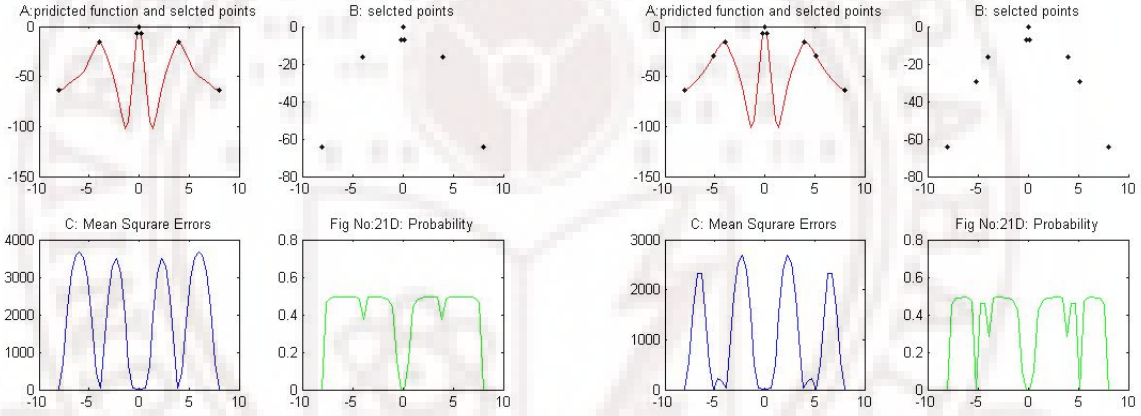
For the first iteration addition of sample points by considering probability and probability with symmetric points is shown in Fig(7.3) and by considering MSE and MSE with symmetric points is as shown in Fig(7.4).Global maximum for rastringin‘s function is obtainable for the second iteration for all the four algorithms.

For the first iteration the maximum probability for the rastringin function is 0.4445 at the



(a) MSE 1st iteration (b) MSE with symmetric 1st iteration

Figure 7.4: Comparing MSE and MSE with Symmetric points for 1st iteration for Rastringin function



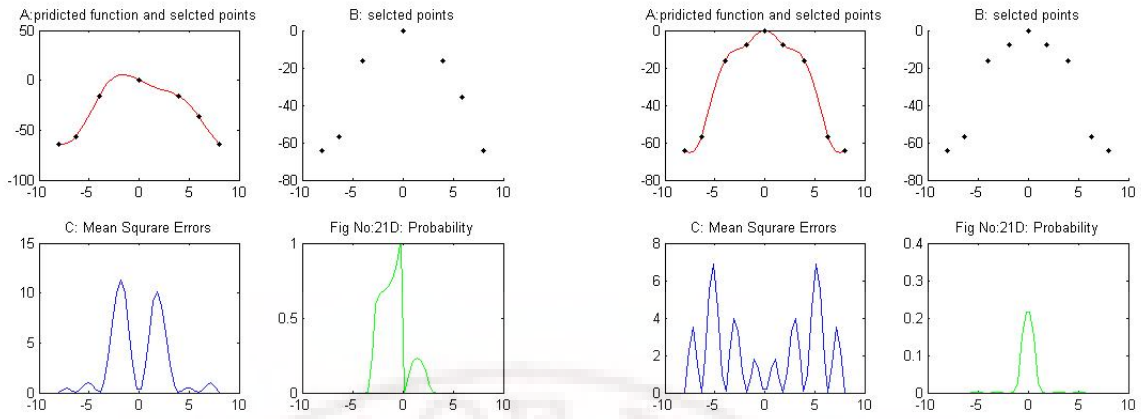
(a) EGO 2nd iteration (b) extended EGO 2nd iteration

Figure 7.5: Comparing EGO and Extended EGO for 2nd iteration for Rastringin function

point x=-0.2051.Conventional EGO adds a sample point at x=-0.2051 and extended EGO algorithm adds 2 points at x=-0.2051 and x=0.2051. Maximum MSE for the first iteration is 15.4527 at the point x=-6.3590. So, MSE algorithm adds sample point at x=-6.3590 and MSE with symmetric points adds two sample points at x=-6.3590 and x=6.3590 for first iteration.

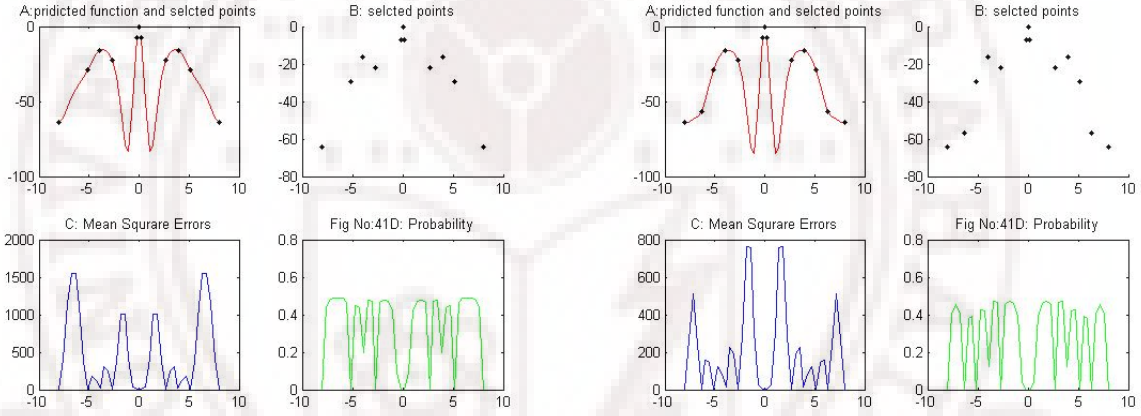
MSE and Probabiliy shown in Fig.(7.3) and Fig.(7.4)are the values after adding the above mentioned sample points in the first iteration. Based on the values of MSE and Probability at the end of the first iteration the next sample point to be added in second iteration is determined.

At the end of the first iteration the maximum probability of rastringin function in



(a) MSE 2nd iteration (b) MSE with symmetric 2nd iteration

Figure 7.6: Comparing MSE and MSE with Symmetric points for 2nd iteration for Rastringin function

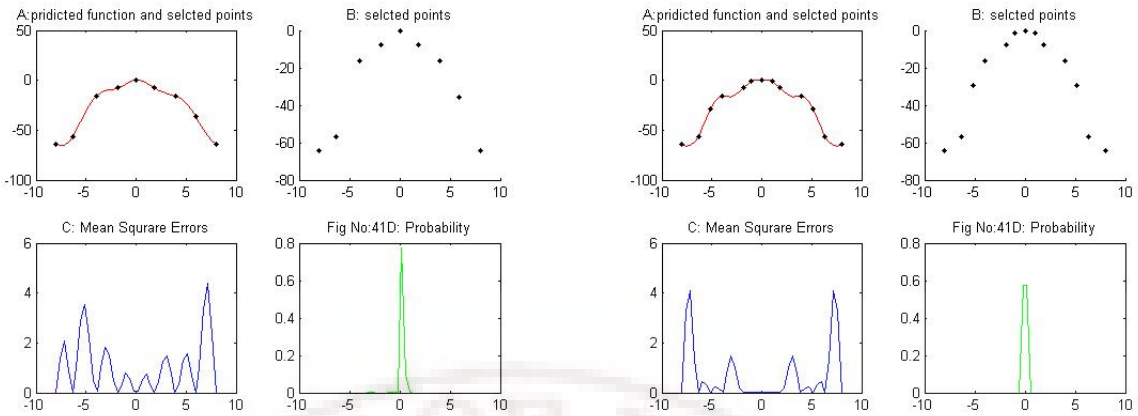


(a) EGO 4th iteration (b) extended EGO 4th iteration

Figure 7.7: Comparing EGO and Extended EGO for 4th iteration for Rastringin function

case of conventional EGO is 0.8984 at [-0.2051]. So, next sample point is added at [-0.2051].This point has already been added in the first iteration incase of extended EGO algorithm. Maximum probability for extended EGO algorithm at the end of first iteration is 0.4950 at [5.1282]. Again two points are added at [5.1282] and [-5.1282].
Value of maximum MSE at the end of first iteration is 78.5957 at [5.9487] in case of MSE algorithm whereas max MSE is 56.0439 at [-1.8642].The maximum MSE value in case of MSE with symmetric points is less than MSE algorithm. Fig.(7.5) and Fig.(7.6) shows the comparisons of algorithms for the second iteration.

At the end of the third iteration maximum probability for conventional EGO is 0.4928 and point is added at[-2.6667].Maximum probability in case of extended EGO is 0.4854 at [6.3590] and points are added at [-6.3590] [6.3590].Maximum MSE for conventional and



(a) MSE 4th iteration

(b) MSE with symmetric 4th iteration

Figure 7.8: Comparing MSE and MSE with Symmetric points for 4th iteration for Rastringin function

extended EGO are 2.6901e+003 and 1.1003e+003 respectively at the end of third iteration. We can observe that MSE for extended EGO is less when compared to conventional EGO.

Maximum MSE for MSE algorithm is 38.8613 at [1.8462] and MSE with symmetric points algorithm is 1.7276 at [1.0256] and points are added at [1.0256], [-1.0256] at the end of third iteration. Fig.(7.7) and Fig.(7.8) shows comparisons for the fourth iteration. Though global optimum is obtainable at the second iteration itself for all the four algorithms for rastringin's function we can observe that MSE with symmetric points algorithm has the lowest MSE.Extended EGO algorithm also has lower MSE value when compared to conventional EGO algorithm. While MSE with symmetric points algorithm gives good prediction of the given function in fewer iteration, extended EGO algorithm tends to add points which converges to global optimum faster.

7.2 Two-dimensional functions

This section demonstrates the performance of extended EGO algorithm for two-dimensional functions. For a two-dimensional function $2^2 = 4$ symmetric points are added at the end of each iteration for extended EGO algorithm. These points are symmetric about x-axis ,y-axis and origin. GUI for extended EGO algorithm is as shown in fig(7.9).In the GUI initially we select the function to be tested. Then enter the initial sampling rate either 2,4 or 8.Then we select the type of algorithm we want to apply on the given function. Let us see the results of the extended EGO algorithm with symmetric points when compared to the conventional EGO algorithm in case of a benchmark optimization test function called Goldstein-price's function. Test area for all the test functions is restricted to square

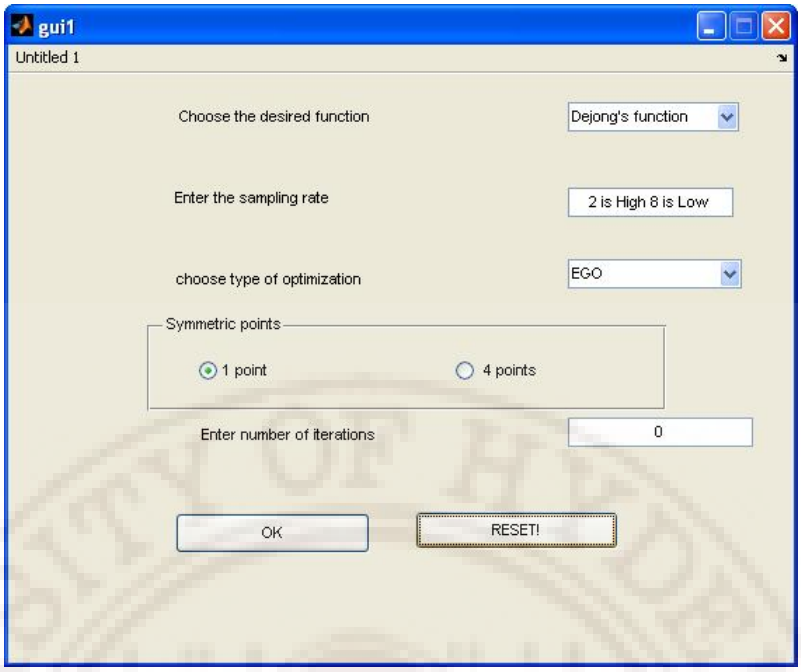


Figure 7.9: Figure showing GUI

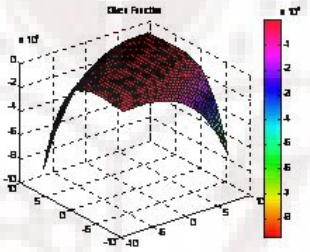


Figure 7.10: Figure of Goldstein prices function

$-8 \leq x \leq 8$ and $-8 \leq y \leq 8$ in 2D plane.

7.2.1 Goldstein-price‘s function

The Goldstein-price function is a global optimization test function. It has only two variables and following definition

$$\begin{aligned} f(x_1, x_2) &= [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \\ &\quad +[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)] \end{aligned}$$

Its global maximum is obtainable at -9.4678 at [-0.2051 -1.0256] and minimum is -8732850 at [-8 8] Initially Goldstein-price‘s function is selected and extended EGO algorithm is run on that function for sample rate of 4 (i.e., 25 points) for 4 iteration as shown in Fig.(7.11) For initial sample rate of four 25 points are selected in case of two-dimensional function in domain -8 to 8 like [-8 -8],[-8 -4],[-8 0],[-8 4],[-8 8],....,[8 -8],[8 -4],[8 0],[8 4],[8 8].The

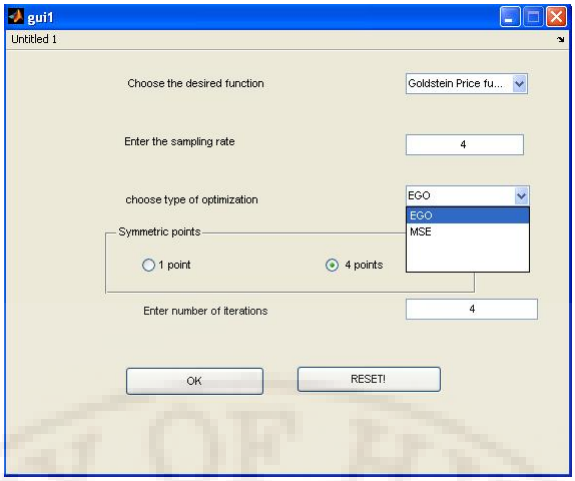


Figure 7.11: Figure showing GUI when Goldstein price function is selected

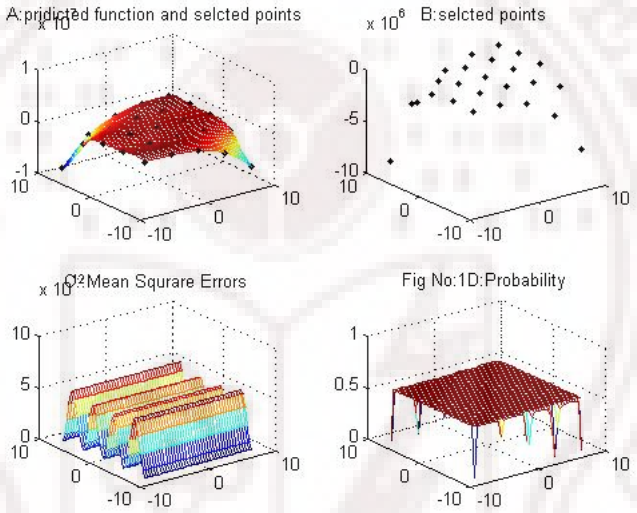


Figure 7.12: Initial sample points and predicted function

initial sample points and predicted function for those sample points along with MSE and Probability for this sample points is shown in fig.(7.12)

With the help of these initial sample points of 25 a DACE model is built based on kriging approximations. DACE model helps in predicting the outcome of 1600 points in the grid generated using grid sampling for the Goldstein-price‘s function. The DACE model created for the test function and its parameters is shown in fig.(7.13)

The value of maximum probability for first iteration is 0.5000 at [5.9487 4.3077] and value of maximum MSE is 5.3235e+012 at [-6.7692 5.9487].Conventional EGO adds point at [5.9487 4.3077] and extended EGO adds four points at [-5.9487 -4.3077], [-5.9487 4.3077], [5.9487 4.3077], [5.9487 -4.3077]. Similarly MSE algorithm adds next sample point at [-6.7692 5.9487] and MSE with symmetric points algorithm adds points at [6.7692 -5.9487],

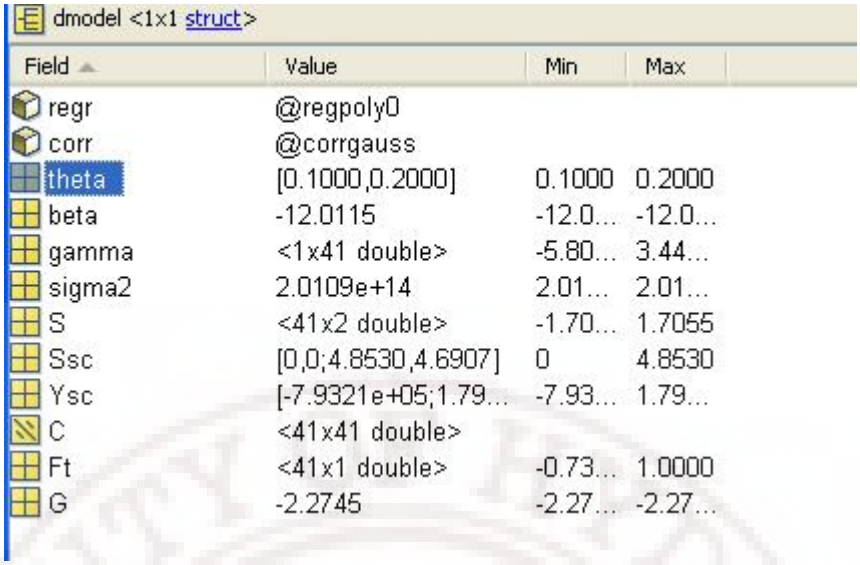


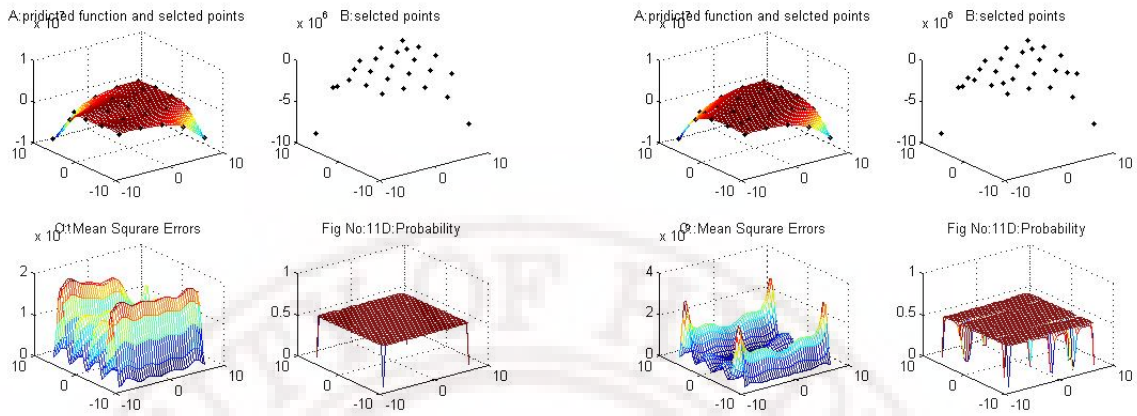
Figure 7.13: Figure on DACE model and its parameters

[6.7692 5.9487], [-6.7692 5.9487], [-6.7692 -5.9487] for the first iteration. The selected points along with the values of Probability and MSE at the end of first iteration are shown in fig.(7.14) and fig.(7.15) for conventional EGO , Extended EGO , MSE and MSE with symmetric points respectively.

At the end of the first iteration maximum probability in case of conventional EGO algorithm is 0.5000 at [0.6154 0.2051] and in case of extended EGO algorithm is 0.5000 at [-0.2051 -0.2051].The points added for second iteration in case of extended EGO algorithm are [0.2051 0.2051], [0.2051 -0.2051], [-0.2051 -0.2051], [-0.2051 0.2051] as shown in Fig.(7.16). The value of maximum MSE at the end of first iteration for MSE algorithm is 1.4818e+011 and point is added at [8.0000 -6.3590].Maximum value of MSE for MSE with symmetric points algorithm at the and of first iteration is 1.2095e+010 at [0.2051 -6.3590].The points added for second iteration are [-0.2051 6.3590], [-0.2051 -6.3590], [0.2051 -6.3590], [0.2051 6.3590] as shown in Fig.(7.17).

At the end of the second iteration maximum probability in case of conventional EGO algorithm is 0.5000 at [-5.1282 -4.3077] and in case of extended EGO algorithm is 0.5252 at [0.2051 -0.6154].The points added for third iteration in case of extended EGO algorithm are [-0.2051 0.6154], [-0.2051 -0.6154], [0.2051 -0.6154], [0.2051 0.6154] as shown in Fig.(7.18). The value of maximum MSE at the end of second iteration for MSE algorithm is 4.7639e+012 and point is added at [-8.0000 -5.9487].Maximum value of MSE for MSE with symmetric points algorithm at the end of second iteration is 1.6316e+012 at [-0.2051 1.8462].The points added for third iteration are [0.2051 -1.8462], [0.2051 1.8462], [-0.2051 1.8462], [-0.2051 -1.8462] as shown in Fig.(7.19).

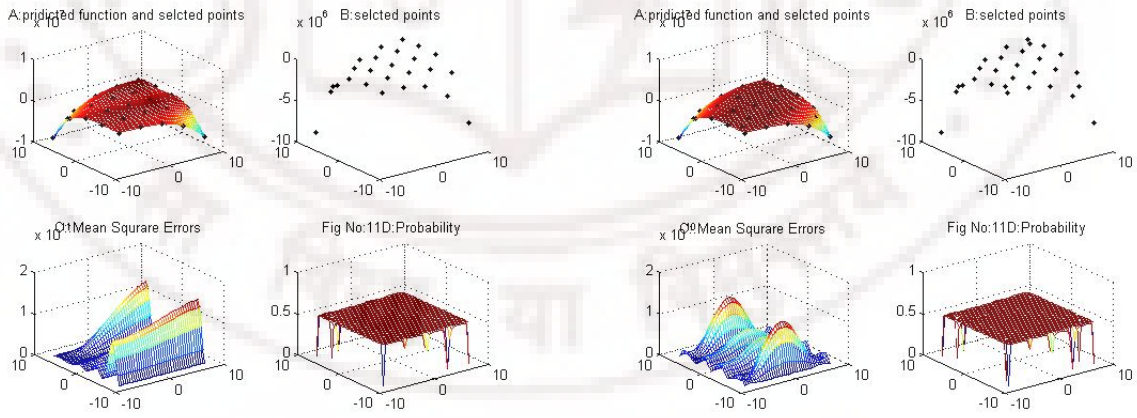
At the end of the third iteration maximum probability in case of conventional EGO



(a) EGO 1st iteration

(b) extended EGO 1st iteration

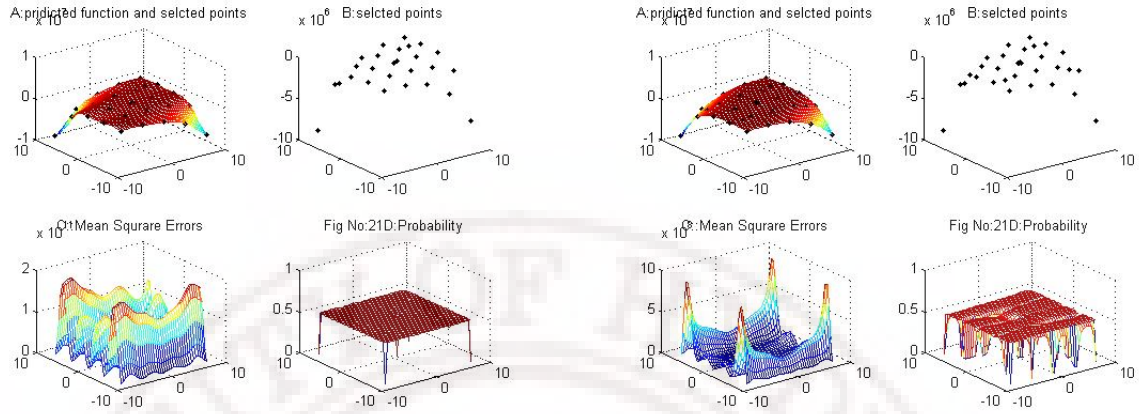
Figure 7.14: Comparing EGO and EGO with Symmetric points for 1st iteration for Goldstein price function



(a) MSE 1st iteration

(b) MSE with symmetric 1st iteration

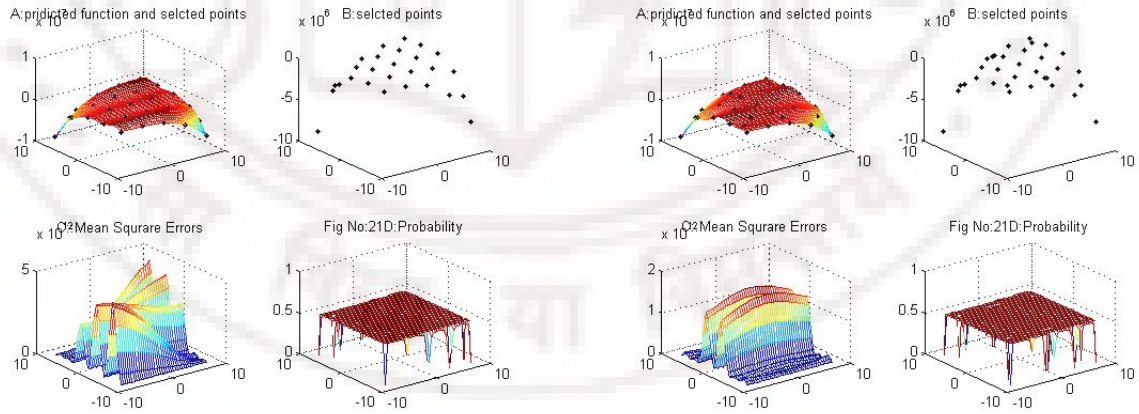
Figure 7.15: Comparing MSE and MSE with Symmetric points for 1st iteration for Goldstein Price function



(a) EGO 2nd iteration

(b) extended EGO 2nd iteration

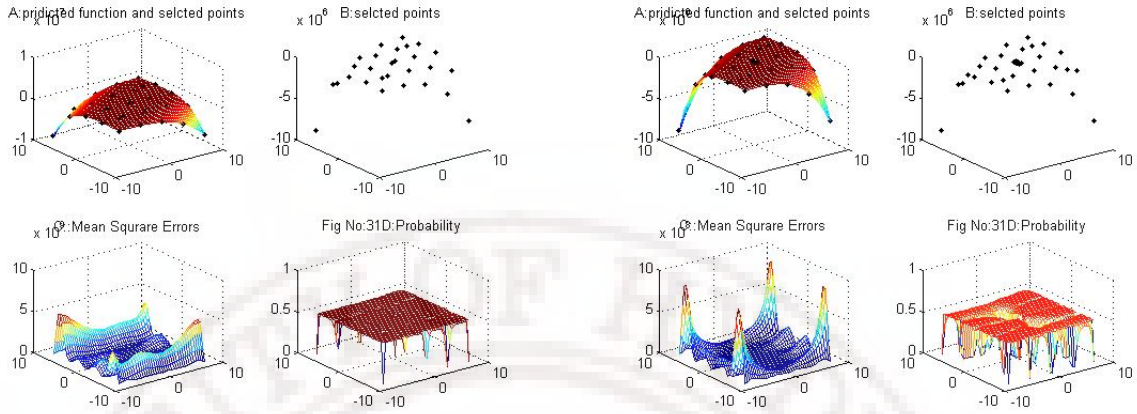
Figure 7.16: Comparing EGO and Extended EGO for 2nd iteration for Goldstein Price function



(a) MSE 2nd iteration

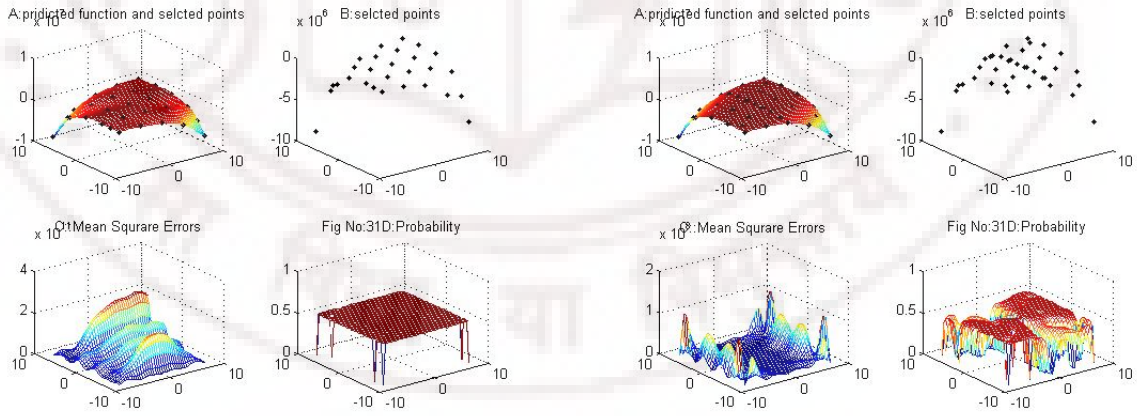
(b) MSE with symmetric 2nd iteration

Figure 7.17: Comparing MSE and MSE with Symmetric points for 2nd iteration for Goldstein Price function



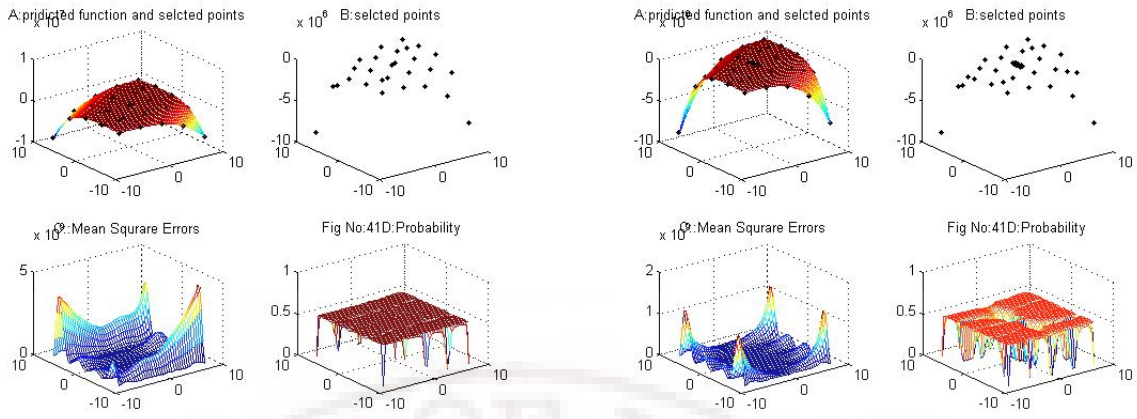
(a) EGO 3rd iteration (b) extended EGO 3rd iteration

Figure 7.18: Comparing EGO and Extended EGO for 3rd iteration for Goldstein Price function



(a) MSE 3rd iteration (b) MSE with symmetric 3rd iteration

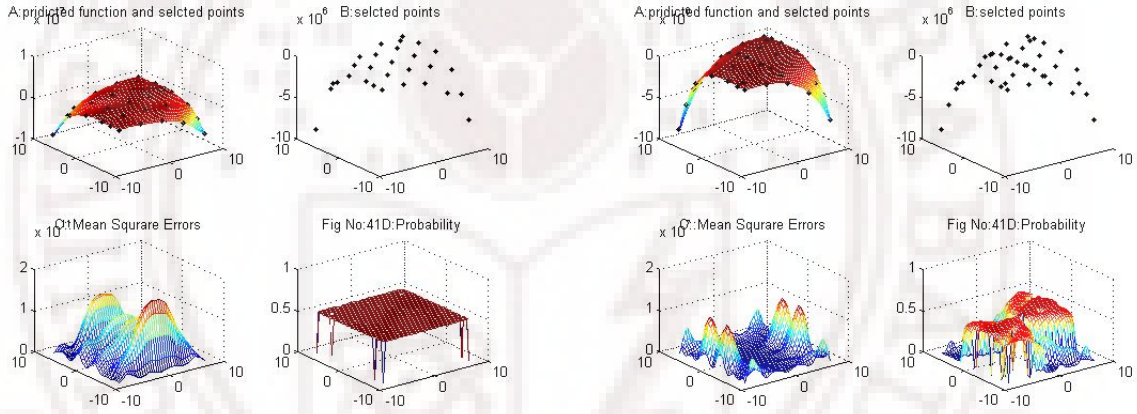
Figure 7.19: Comparing MSE and MSE with Symmetric points for 3rd iteration for Goldstein Price function



(a) EGO 4th iteration

(b) extended EGO 4th iteration

Figure 7.20: Comparing EGO and Extended EGO for 4th iteration for Goldstein Price function



(a) MSE 4th iteration

(b) MSE with symmetric 4th iteration

Figure 7.21: Comparing MSE and MSE with Symmetric points for 4th iteration for Goldstein Price function

algorithm is 0.5001 at [-0.2051 -0.2051] and in case of extended EGO algorithm is 0.5880 at [-0.2051 -1.0256].The points added for fourth iteration in case of extended EGO algorithm are [0.2051 1.0256], [0.2051 -1.0256], [-0.2051 -1.0256], [-0.2051 1.0256] as shown in Fig.(7.20). The value of maximum MSE at the end of third iteration for MSE algorithm is 2.2424e+011 and point is added at [6.3590 6.3590].Maximum value of MSE for MSE with symmetric points algorithm at the and of third iteration is 1.0446e+008 at [-8.0000 -6.3590].The points added for fouth iteration are [8.0000 6.3590], [8.0000 -6.3590], [-8.0000 -6.3590], [-8.0000 6.3590] as shown in Fig.(7.21).

The result of the function after four iterations for extended EGO algorithm,, the predicted function and original function with the selected points, the associated MSE and probability is shown in fig.(7.22)

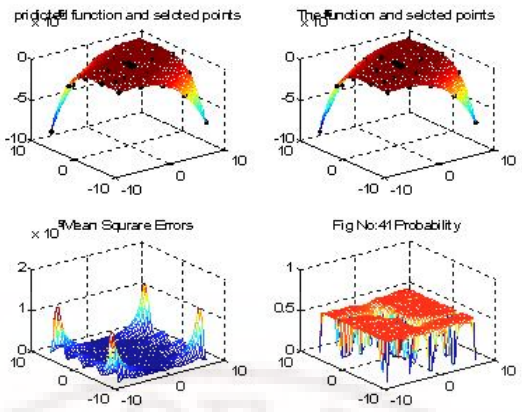


Figure 7.22: Predicted function,given function,MSE and probability of Goldstein Price function for Extended EGO algorithm

The maximum mean square error in this case is 1.1797e+009 and maximum probability is 0.5928.
While the extended EGO algorithm is able to converge to global maximum in 4 iterations, the conventional EGO algorithm fails to converge to global maximum and settles at some optimum in 7th iteration. Also, the mean square error associated with conventional EGO algorithm after 7 iterations is 4.3252e+009 which is convincingly more than the extended EGO algorithm in the fourth iteration itself.
It is observed that maximum MSE is not a good criterion for choosing the next sample point because the point of maximum MSE implies that the error associated with the predicted value of the function is high. This need not point which converges to the global maximum. Hence, both the cases of MSE and MSE with symmetric points fails to converge to global maximum in first iteration itself with more error rate than the conventional EGO algorithm which also failed to converge to global maximum in 7th iteration.
These different cases are run for different sampling rates of 2, 4 and 8.Higher sample rate implies more number of points and lesser sample rate implies less number of points. The result of comparisons is as shown in the table:7.1

The table(7.1) shows efficiency of the extended EGO algorithm over conventional EGO algorithm. This table shows four different metrics of comparison. Initially cost associated with different algorithms
Cost is calculated as

$$\begin{aligned} Cost &= \text{initial sample points} + \text{number of iterations take to converge} \\ &\quad *(\text{additional sample points taken for each iteration}) \end{aligned}$$

Second metric of comparison is number of evaluations taken to converge to global maximum

	sample2(81 points)				sample4(25 points)				sample8(9 points)			
	cost	Eval	Error	MSE	cost	Eval	Error	MSE	cost	Eval	Error	MSE
EGO	84*	3^{rd}	230.2%	3.7567e+008	32*	7^{th}	287.5%	43252e+009	12*	3^{rd}	326.7%	1.3230e+012
EGO symmetric	93	3^{rd}	—	1.0079e+008	41	4^{th}	—	1.1797e+009	29	5^{th}	—	8.5366e+010
MSE	82*	1^{st}	428.1%	5.7785e+008	26*	1^{st}	428.1%	1.4818e+011	10*	1^{st}	428.1%	2.3627e+012
MSE symmetric	85*	1^{st}	428.1%	4.0972e+007	29*	1^{st}	428.1%	1.2095e+010	13*	1^{st}	428.1%	2.3627e+012

Table 7.1: Comparison of different algorithms for Goldstein price's function

which is abbreviated as "Eval".

The third metric of comparison is the error rate. Error rate is calculated in cases where the algorithm fails to converge to global optimum and gets trapped at some point. Error rate can be defined as the percentage error by which the algorithm has failed to reach global optimum. Percentage error is calculated as

$$\text{Percentage of error} = \frac{\text{(Global Optimum-Converged Optimum)}}{\text{Global Optimum}} * 100 \tag{7.2}$$

The final metric of comparison is MSE. As defined earlier MSE is the measure of accuracy of the predicted value.

In Table(7.1) the presence of * in cost column implies that the algorithm has failed to converge to the global maximum and gets trapped at some point for some number of Evaluations as specified by Eval column. For these columns there exists percentage error at which it has converged. The point of convergence and converged optimum in these cases is shown in Table(7.2).

	sample2(81 points)		sample4(25 points)		sample8(9 points)	
	converged optimum	point	converged optimum	point	converged optimum	point
EGO	-31.2717	[1.8462 0.2051]	-36.6895	[-0.2051 1.0256]	-40.4060	[-0.2051 -0.2051]
EGO symmetric	—	—	—	—-	—	—
MSE	-50	[0 0]	-50	[0 0]	-50	[0 0]
MSE symmetric	-50	[0 0]	-50	[0 0]	-50	[0 0]

Table 7.2: Converged optimum and point of convergence for Goldstein-Price's function

7.2.2 Branin's function

Branin function is a global optimization test function having only 2 variables. The function has the following definition

$$f(x_1, x_2) = a(x_2 - bx_1^2 + cx_1 - d)^2 + e(1 - f)cos(x_1) + e \tag{7.3}$$

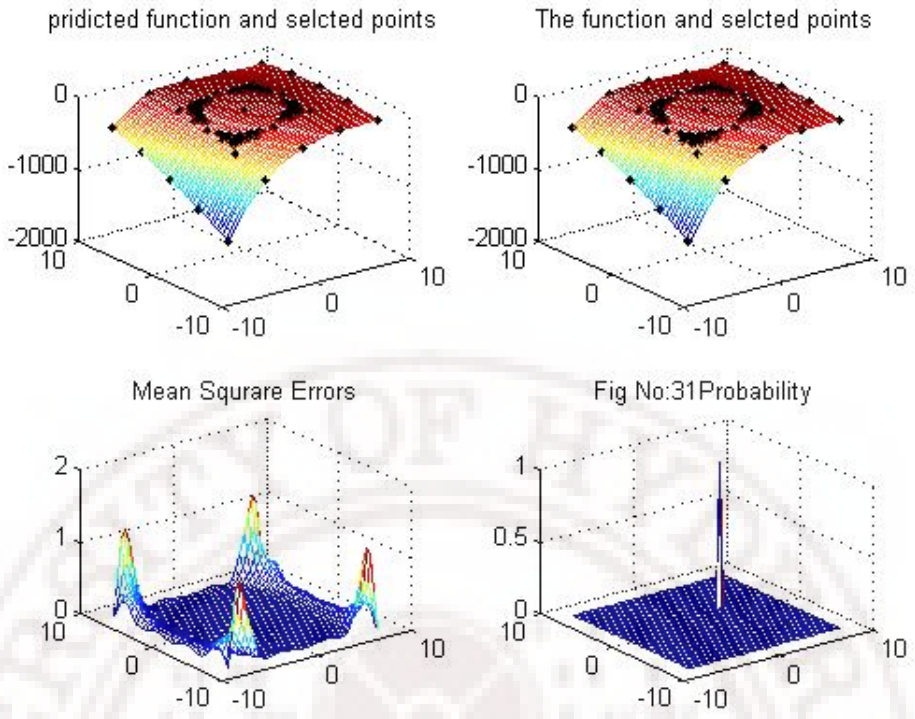


Figure 7.23: Figure showing result of Branin's function

It is recommended to set the following values of parameters a=1, b=$\frac{5.1}{4\pi^2}$, c=$\frac{5}{\pi}$,d=6,e=10,f=$\frac{1}{8\pi}$. Its global maximum is -0.4228 obtainable at [3.0769 2.2564] and minimum is -1.2336e+003 at [-8 -8].

In case of Branin's function our extended EGO algorithm outperforms conventional EGO

	sample2(81 points)				sample4(25 points)				sample8(9 points)			
	cost	Eval	Error	MSE	cost	Eval	Error	MSE	cost	Eval	Error	MSE
EGO	82	1^{st}	—	0.1845	28*	3^{rd}	53.3%	269.1168	11*	2^{nd}	1937%	9.3311e+004
EGO symmetric	85	1^{st}	—	0.0032	37	3^{rd}	—	5.8409	11*	2^{nd}	1937 %	4.8138e+004
MSE	82*	1^{st}	801.9%	0.8909	53*	28^{th}	865%	0.0155	21*	12^{th}	1472.2%	60.0872
MSE symmetric	85*	1^{st}	801.9%	0.0667	101	19^{th}	—	2.7387e-007	105*	24^{th}	132.2%	4.0619e-007

Table 7.3: Comparison of different algorithms for Branin's function

algorithm for sample rate-2 and sample rate-4 but fails for sample rate-8. Though it fails in the case of higher sample rate which implies less number of sample points the error rate associated with it same as in the case of conventional EGO algorithm as shown in Table(7.3). Point of convergence is same in both the algorithms as shown in Table(7.4). We can observe that the MSE associated with conventional EGO is 9.3311e+004 which is more than the MSE value of our extended EGO algorithm 4.8138e+004 for the same

number of function evaluations. MSE with symmetric points algorithm succeeds in converging to optimum only for sample rate of 4 at 19th iteration and fails in all other cases. Fig.(7.23) shows the result of Branin's function for extended EGO algorithm for sample rate of 4 where it got converged to global maximum at third iteration.

	sample2(81 points)		sample4(25 points)		sample8(9 points)	
	converged optimum	point	converged optimum	point	converged optimum	point
EGO	—	—	-0.6482	[3.0769 1.8462]	-8.6128	[8.0 1.4359]
EGO symmetric	—	—	—	—-	-8.6128	[8.0 1.4359]
MSE	-3.8132	[4 2]	-4.0803	[2.2564 2.6667]	-6.6476	[2.2564 4.7179]
MSE symmetric	-3.8132	[4 2]	—	—	-0.9819	[3.0769 3.0769]

Table 7.4: Converged optimum and point of convergence for Branin's function

7.2.3 Six-hump camel back function

It is a global optimization test function with two variables and following definition:

$$f(x_1, x_2) = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2 \tag{7.4}$$

Its global maximum is 0.9 obtainable at [0.2051 -0.6154] and minimum is -95228 at [-8 -8] and [8 8].

	sample2(81 points)				sample4(25 points)				sample8(9 points)			
	cost	Eval	Error	MSE	cost	Eval	Error	MSE	cost	Eval	Error	MSE
EGO	84	3^{rd}	230.2%	5.3697e+004	31	6^{th}	—	8.6516e+006	17*	8^{th}	90.2%	3.3791e+007
EGO symmetric	89	2^{nd}	—	1.9047	45	5^{th}	—	0.0219	29	5^{th}	—	4.5842e+007
MSE	82*	1^{st}	100%	8.1227e+004	26*	1^{st}	100%	8.7205e-004	10*	1^{st}	100%	1.2597e+009
MSE symmetric	85*	1^{st}	100%	8.8060e+003	29*	1^{st}	100%	6.0318e+005	13*	1^{st}	100%	1.2597e+009

Table 7.5: Comparison of different algorithms for six hump camel back function

In Six hump camel back function our Extended EGO algorithm performs better in all cases compared to all other algorithm as we can see in Table(7.5).Extended EGO got converged to global maximum in less number of evaluations for all sampling rates while conventional EGO algorithm failed to converge to optimum for sample rates 2 and 8.It got converged to global maximum only at sample rate of four at 6th iteration. MSE and MSE with symmetric points algorithm failed to converge in all the cases for the first iteration itself. Table(7.6) shows the point of convergence and converged optimum for six hump camel back function.

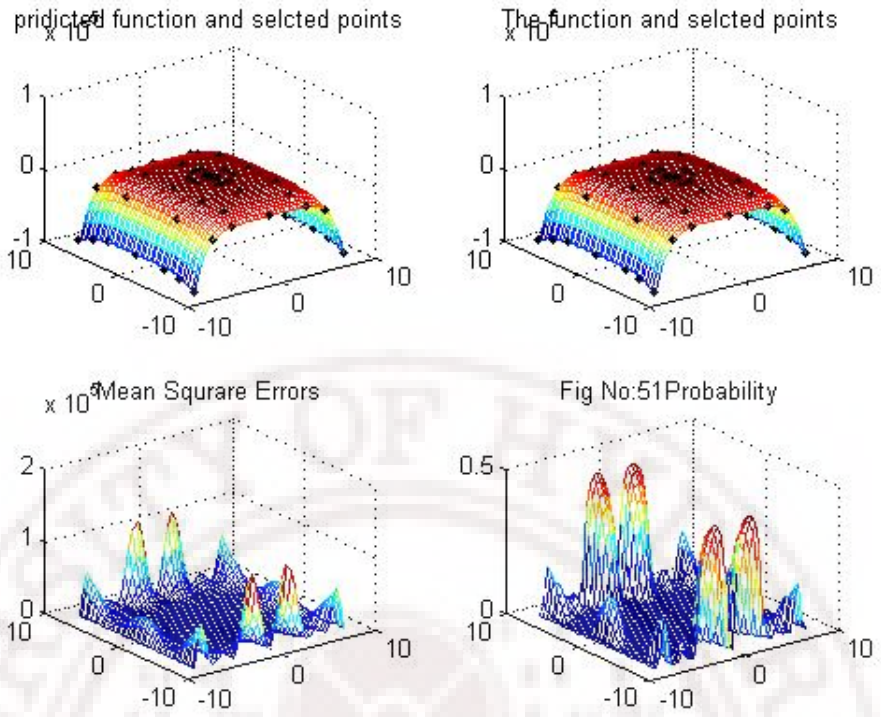


Figure 7.24: Figure of Six hump camel back function

	sample2(81 points)		sample4(25 points)		sample8(9 points)	
	converged optimum	point	converged optimum	point	converged optimum	point
EGO	—	–	—	—	0.0881	[-0.6154 0.6154]
EGO symmetric	—	—	—	—-	—	—
MSE	0	[0 0]	0	[0 0]	0	[0 0]
MSE symmetric	0	[0 0]	0	[0 0]	0	[0 0]

Table 7.6: Converged optimum and point of convergence for six hump camel back function

7.2.4 Rosenbrock's valley function

It is a global optimization test problem also known as banana function. The global optimum lies inside a long, narrow, parabolic shaped flat valley. To find the valley is trivial, however convergence to the global optimum is difficult and hence this problem has been frequently used for testing the performance of the optimization algorithms. Function has the following definition

$$f(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right] \tag{7.5}$$

Test area is restricted the square $-8 \leq x \leq 8$ and $-8 \leq y \leq 8$.Its global maximum ymax=-0.0698 is obtainable at [1.0256 1.0256] and minimum is -518481 at [-8 -8].

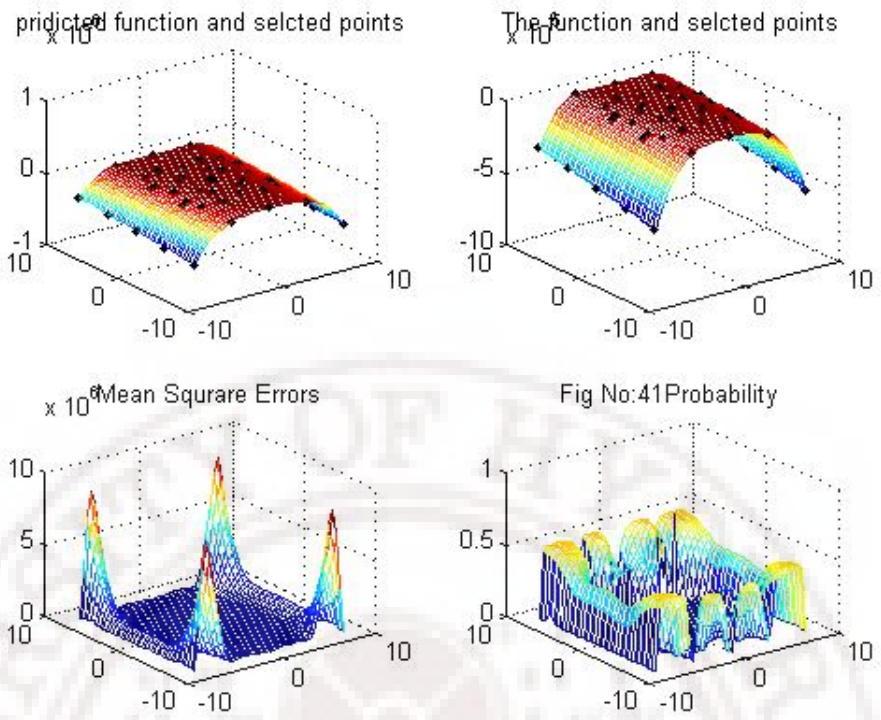


Figure 7.25: Figure of Rosenbrock's valley function

In case of Rosenbrock's valley function both conventional EGO and Extended EGO

	sample2(81 points)				sample4(25 points)				sample8(9 points)			
	cost	Eval	Error	MSE	cost	Eval	Error	MSE	cost	Eval	Error	MSE
EGO	85	4^{th}	—	224640	39	14^{th}	—	56973000	31	22^{nd}	—	1.9600e+008
EGO symmetric	89	2^{nd}	—	11706	41	4^{th}	—	26386000	49	10^{th}	—	9541700
MSE	82*	1^{st}	39.6%	130580	26*	1^{st}	39.6%	177750000	10*	1^{st}	39.6%	5.7525e+008
MSE symmetric	85*	1^{st}	39.6%	48628	29*	1^{st}	39.6%	1.1414e+010	13*	1^{st}	39.6%	5.7525e+008

Table 7.7: Comparison of different algorithms for Rosenbrock's valley function

succeeded in converging to optimum while extended EGO has converged to optimum in less number of evaluations than conventional EGO as shown in Table(7.7). MSE and MSE with symmetric points algorithm failed in all cases. They got trapped to maximum y=-1 at[0 0] with error rate of 39.6%. Fig(7.25) shows the result of our extended EGO algorithm for sample rate of four after four iterations for Rosenbrock's valley function.

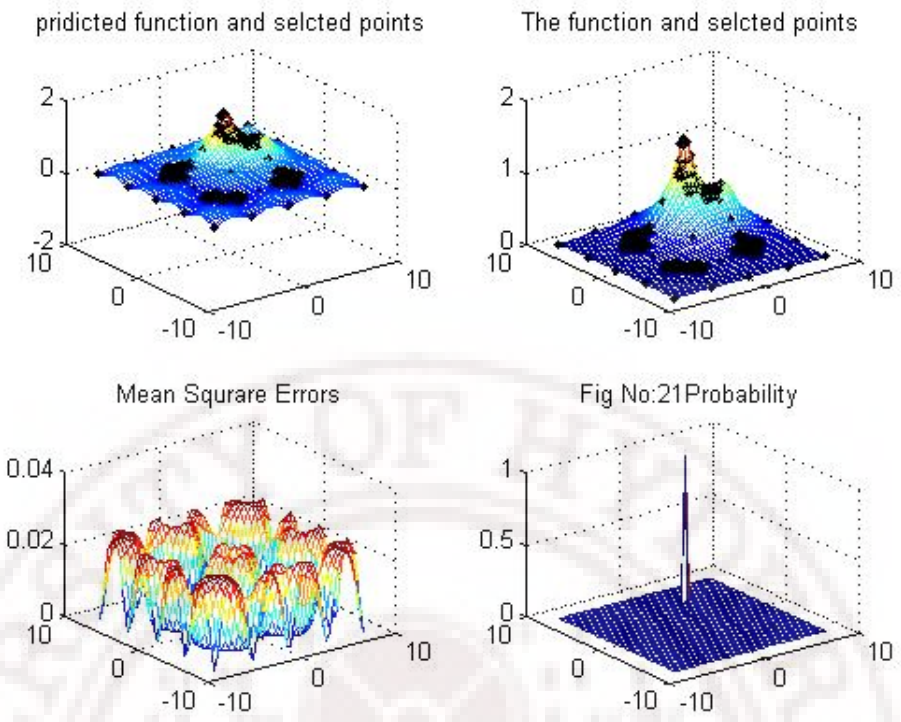


Figure 7.26: Figure of shekel's foxholes function

7.2.5 Shekel's foxholes function

This is a multimodal test function. It has the following definition

$$f(x) = -\sum_{i=1}^{m}\left(\sum_{j=1}^{n}\left[(x_j - a_{jj})^2 + c_j\right]\right)^{-1} \tag{7.6}$$

Where $(c_i, i = 1, \cdots, \cdots, m)(a_{ij}, j = 1, \cdots, \cdots, n, i = 1, \cdots, \cdots, m)$ are constant numbers fixed in advance.It is recommended to set m=30. Its global maximum is 1.2476 obtainable at [3.0769 5.1282] and minimum is 0.0188 at [-8 -8].

In case shekel's foxholes function for sample rate of 8 extended EGO algorithm converges

	sample2(81 points)				sample4(25 points)				sample8(9 points)			
	cost	Eval	Error	MSE	cost	Eval	Error	MSE	cost	Eval	Error	MSE
EGO	84	3^{rd}	—	0.0198	27	2^{nd}	—	0.0589	12*	3^{rd}	72.3 %	0.0111
EGO symmetric	93	3^{rd}	—	0.0159	33	2^{nd}	—	0.0319	21	3^{rd}	—	0.0660
MSE	126	45^{th}	—	0.0046	58*	33^{rd}	26.64%	0.0025	69*	60^{th}	15.13%	0.0055
MSE symmetric	117*	9^{th}	2.1%	0.0043	2005*	45^{th}	16.86%	1.0021e-004	281*	68^{th}	2.1%	2.7933e-004

Table 7.8: Comparison of different algorithms for shekel's foxholes function

to global maximum in 3rd iteration while conventional EGO algorithm gets trapped at some

point for 3rd iteration. For sample rates of 2 and 4 both extended EGO and Conventional EGO takes same number of evaluations in converging to optimum but MSE value is less for Extended EGO than conventional EGO as shown in Table(7.8).MSE and MSE with symmetric points algorithm fails to converge to optimum and gets trapped at some point as shown in Table(7.9).

	sample2(81 points)		sample4(25 points)		sample8(9 points)	
	converged optimum	point	converged optimum	point	converged optimum	point
EGO	—	—	—	—	0.3449	[6.7692 8.000]
EGO symmetric	—	—	—	—	—	—
MSE	—	—	0.9152	[2.2564 4.3077]	1.0121	[3.0769 5.5385]
MSE symmetric	1.2214	[3.0769 4.7179]	1.0372	[3.4872 4.7179]	1.2214	[3.0769 4.7179]

Table 7.9: Converged optimum and point of convergence for shekel's foxholes function

Discussion

From all the above results we can conclude that our extended EGO algorithm has given better results in terms of number of evaluations taken to converge to optimum and also MSE.MSE is not considered to be good criteria when the objective is to find global optimum but a less value of MSE certainly implies that the approximate function closely resembles the original function. Though the cost associated with our extended algorithm is more when compared to conventional EGO algorithm it takes less number of iterations to converge to optimum and it is likely certain to converge to optimum in many cases where conventional EGO algorithm and MSE algorithms fails to converge to optimum. Even in cases where extended EGO algorithm fails to converge to optimum, like for Branin's function with sample rate 8 it failed in finding global optimum, its error rate is same as conventional EGO and MSE value is less than conventional EGO. When extended EGO algorithm is deployed in Parallel computing systems where job is simultaneously submitted to the parallel machines then the cost associated with it will also be sliced. So, for a huge job where our objective is reaching optimum quickly in few iterations with the help of unbiased surrogate model our Extended EGO algorithm proves to be beneficial.

Chapter 8

Conclusion

In this report we have surveyed the existing methodologies for surrogate modeling and global optimization using model based optimization techniques. We have studied the theory associated with kriging and DACE and chosen it as our surrogate model. Also different criteria for adding sample points in successive iterations are studied. DACE is used for optimization purpose in EGO. Although EGO performs best in providing models for optimization, it would be desirable to obtain results under all possible designs. Iterative evaluation of DACE model by adding sample points in search of optimum tends to lose its symmetric nature. Also, conventional EGO has drawback of adding sample points near previously found optimum ignoring other design sites. Hence, the concept of symmetric points has been introduced in EGO which we call as extended EGO which would serve as an enhanced surrogate model to get an unbiased global optimization .Different benchmark test functions of optimization have been collected and the efficiency of extended EGO algorithm is tested in 2D plane.

Extended EGO algorithm is implemented along with several criteria for adding sample points .It is observed from the results that our extended EGO algorithm outperforms all the other algorithms in terms of number of evaluations taken to converge to optimum. It is more certain to converge to optimum than conventional EGO algorithm. It is also observed that neither MSE nor MSE with symmetric point is a good criterion for adaptive sampling when the objective is converging to global optimum. We can say that MSE with symmetric points is a good criterion when the objective is to produce a good approximate model for the given function. Thus an unbiased more certain surrogate model is developed for finding global optimum using extended EGO algorithm.

Further this extended EGO algorithm can also be applied to multi-objective optimization.

Bibliography

[1] Global optimization. http://en.wikipedia.org/wiki/Globaloptimization.

[2] T.Wagner D. Beirmann, k.Weinert. Model based optimization revisited: Towards real word processes. *IEEE congress on Evolutionary computation (CEC 2008)*, pages 2975–2982.

[3] L.F.P Etman. Design and analysis of computer experiments: The method of sacks et al. Engineering Mechanics report WFW 94.098, 1994.

[4] Amitay Isaacs. *Direct-search methods and DACE*. PhD thesis, Department of Aerospace Engineering , Indian Institute of Technology Bombay, May 2003.

[5] T.J. Mitchell H.P. Wynn J. Sacks, W.J. Welch. Design and analysis of computer experiments. 4(4):409–435, 1989.

[6] Schonlau M. Welch W.J. Jones, D.R. Efficient global optimization of expensive black-box functions. *Journal of global optimization*, 13(4):455–492, 1998.

[7] Runze Li Kai-Tai Fang and Agus Sudjianto. *Design and Modeling for Computer Experiments*. Computer Science and Data Analysis Series. Published by CRC Press, 2006.

[8] Colin H. Hansen David J. Murphy Rick C. Morgans, Anthony C. Zandder. Ego shape optimization of horn-loaded loudspeakers. *Optim Eng*, (9):361–374, Nove 2007.

[9] Papalambros P.Y. Sasena, M.J. and P. Goovaerts. Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering Optimization*, 3(34):263–278.

[10] Parkinson M. Goovaerts P. Papalamabros P. Reed M. Sasena, M.J. Adaptive experimental design applied to an ergonomics testing procedure. in the Proceedings of DETC02 ASME 2002 Design Engineering Technical Conferences and Computers and Information in Engineering Conference Montreal, Canada, 2002.

[11] M. Schonlau. *Computer Experiments and Global Optimization*. PhD thesis, University of Waterloo, Waterloo, Canada, 1997.

[12] Booker A.J. Ghosh D. Giunta A.A. Koch P.N. Yang R. Simpson, T.W. Approximation methods in multidisciplinary analysis and optimization: A panel discussion. Approximation Methods Panel, 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, GA, 2002.

[13] Hans Bruun Nielsen Soren N.Lophaven and Jacob Sondergaard. Dace a matlab kriging toolbox. *IMM- Technical University of Denmark*, 2.0, August 1, 2002.

[14] William I. Notz Thomas J. Santner, Brian J. Williams. *The Design and Analysis of Computer Experiments*. Springer series in statistics. 2003.

[15] Thomas weise. Global optimization algorithms theory and application, 2009-05-01. http://www.it-weise.de/projects/book.pdf.

Appendix1

Maximum Likelihood Estimation

The maximum likelihood estimation (MLE) that we shall consider is a parametric form of density estimation problem. Suppose $x_1, x_2, \cdots, x_m$ are independent and identically distributed (iid) with the common probability density function ψ. Suppose ψ is parameterized with respect to θ. The maximum likelihood estimation problem is to find θ such that conditional probability of $x_1, x_2, \cdots, x_m$ given θ, i.e. $P(x_1, x_2, \cdots, x_m \mid \theta)$ is maximum. Using that x_is are iid,

$$\theta_{MLE} = argmax(P(x_1, x_2, \cdots, x_m \mid \theta))$$

Since $x_i s$ are iid,

$$\theta_{MLE} = argmax\left(\prod_1^m p(x_i \mid \theta)\right)$$

$$\theta_{MLE} = argmax\left(\prod_1^m \psi(x_i, \theta)\right)$$

In our case, we frame an MLE problem to find all of θ,σ,β such that the joint probability distribution given by $\prod_1^m \psi\left(z(s_i), \theta, \sigma, \beta\right)$ is maximized, where $z(s_i) = y(s_i) - f(s_i)^T\beta$ is the error at site s_i appearing out of a stochastic process. The kriging model is a combination of multivariate normal model and a linear model. If the stochastic process is taken as Gaussian then the probability density function is given by

$$\prod_1^m \psi\left(z(s_i), \theta, \sigma, \beta\right) = \frac{1}{((2\pi)\sigma^2 det(R))^{m/2)}} exp\left[\frac{-(Y-F\beta)^T R^{-1}(Y-F\beta)}{2\sigma^2}\right]$$

Where R the correlation matrix is function of θ , $Y = [y(s_1), y(s_2), \cdots, y(s_m)]^T$ is vector of outputs at the chosen sites, and $F = [f(s_1), f(s_2), \cdots, f(s_m)]^T$ is an $m \times p$ matrix holding the regression functions evaluated at the chosen sites and β is as defined in Eq.(4.6). Hence MLE problem now is,

$$(\theta, \sigma, \beta)_{MLE} = argmax\left[\frac{1}{((2\pi)\sigma^2 det(R))^{m/2}} exp\left[\frac{-(Y-F\beta)^T R^{-1}(Y-F\beta)}{2\sigma^2}\right]\right]$$

Taking natural logarithms we get log likelihood problem

$$(\theta, \sigma, \beta)_{MLE} = argmax\left[\frac{-1}{2}\left[\text{m } \ln\sigma^2 + \ln \det(\text{R}) + \frac{(Y-F\beta)^T R^{-1}(Y-F\beta)}{\sigma^2}\right]\right]$$

To solve this we must differentiate the log likelihood with respect to θ, σ, and β and put the respective partial derivatives equal to 0.σ and R are independent of β so the MLE of β denoted by β^* is obtained as

$$\frac{\delta}{\delta\beta}\left((Y - F\beta)^T R^{-1}(Y - F\beta)\right) = 0$$

$$\beta^* = (F^T R^{-1} F)^{-1} F^T R^{-1} Y$$

This MLE β^* is the same as the generalized least squares estimate of the regression problem. Similarly, the MLE of σ^2, denoted by σ^{*2} is obtained as

$$\frac{\delta}{\delta\sigma_2}\left[\text{m } \ln\sigma^2 + \frac{(Y - F\beta^*)^T R^{-1}(Y - F\beta^*)}{\sigma^2}\right] = 0$$

$$\sigma^{*2} = \frac{1}{m}\left((Y - F\beta^*)^T R^{-1}(Y - F\beta^*)\right)$$

We see that parameters β and σ^2 are decoupled. In fact it is clear that both σ^{*2} and β^* are both essentially functions of θ. Thus the MLE problem posed above is in fact a problem of finding the MLE of only θ. Although the least squares solution for β and the MLE β^* are identical, this fact could not have been established by finding least squares solution first and then imposing MLE separately on θ.
On solving the MLE for θ, σ^{*2} and β^* get fixed as a consequence. This means that by choosing n critical values in the vector θ, the model and as we shall see in the next section, the predictor can be determined. This remarkable simplicity is attributed to the choice of the stochastic process as Gaussian. MLE for θ is reposed as

$$\theta_{MLE} = argmax\left[\frac{-1}{2}(\text{m } \ln\sigma^{*2} + \ln \det(\text{R})\right]$$

This is an optimization problem that has to be solved numerically. Having solved for θ and having found σ^{*2} and β^*we are in a position now to create a predictor for the function y(x) to predict its value over the domain D.

Appendix2

Deriving the BLUP using the method of Lagrange Multipliers

The BLUP (best linear unbiased predictor) requires that we solve:

1. Minimize MSE=$E(\mid \hat{y}(x) - y(x) \mid^2)$ with respect to c(x)
2. Subject to $E(\hat{y}(x))$ =E(y(x))

The mean square error at untried x is

$$MSE(x) = E\left\{\mid c(x)^T(F\beta^* + Z) - f(x)^T\beta^* - z(x) \mid^2\right\}$$

Where

$$Z = [z(s_1), z(s_2), \cdots, z(s_m)]^T$$

Since E(z)=E(Z)=0 the unbiasedness condition becomes $F^T c(x) = f(x)$

$$MSE(x) = \sigma^2\left(1 + c(x)^T Rc(x) - 2c(x)^T r(x)\right)$$

Where r(x) is a vector that holds the correlations between the untried x and sites in S . $r(x) = [\rho(s_1, x), \rho(s_2, x), \cdots, \rho(s_m, x)]^T$
Thus the constrained optimization problem stated above is to minimize MSE (x) subject to $F^T c(x) = f(x)$. This can be solved using the method of Lagrange multipliers. The Lagrange equation is with a vector Lagrange multipliers λ is

$$L(c, \lambda) = MSE(x) - \lambda^T(F^T c(x) - f(x))$$

$$\frac{\delta L}{\delta \lambda} = 0 \Rightarrow F^T c(x) - f(x) = 0 \& \frac{\delta L}{\delta c} = 0 \Rightarrow 2\sigma^2(Rc(x) - r(x)) - F\lambda = 0$$

$$c(x) = R^{-1}(r(x) + F(F^T R^{-1} F)^{-1}(f(x) - F^T R^{-1} r(x)))$$

Thus the BLUP can now presented as below. Using that R is symmetric,

$$\hat{y}(x) = r(x)^T R^{-1} Y - (F^T R^{-1} r(x) - f(x))^T (F^T R^{-1} F)^{-1} F^T R^{-1} Y$$

MODEL BASED OPTIMIZATION

Presented By

P.SHALINI

MTECH AI

(07MCMI16)

Under Guidance of

Prof.C.R.Rao

Dr.Rajeev wankar

CONTENTS

- SURROGATE MODEL
- DACE
- KRIGIGNG
- INTRODUCTION TO OPTIMIZATION
- EGO
- SUMMARY-FUTURE WORK

SURROGATE MODEL

- Surrogate models are approximation models which replace the behavior of original high fidelity code.

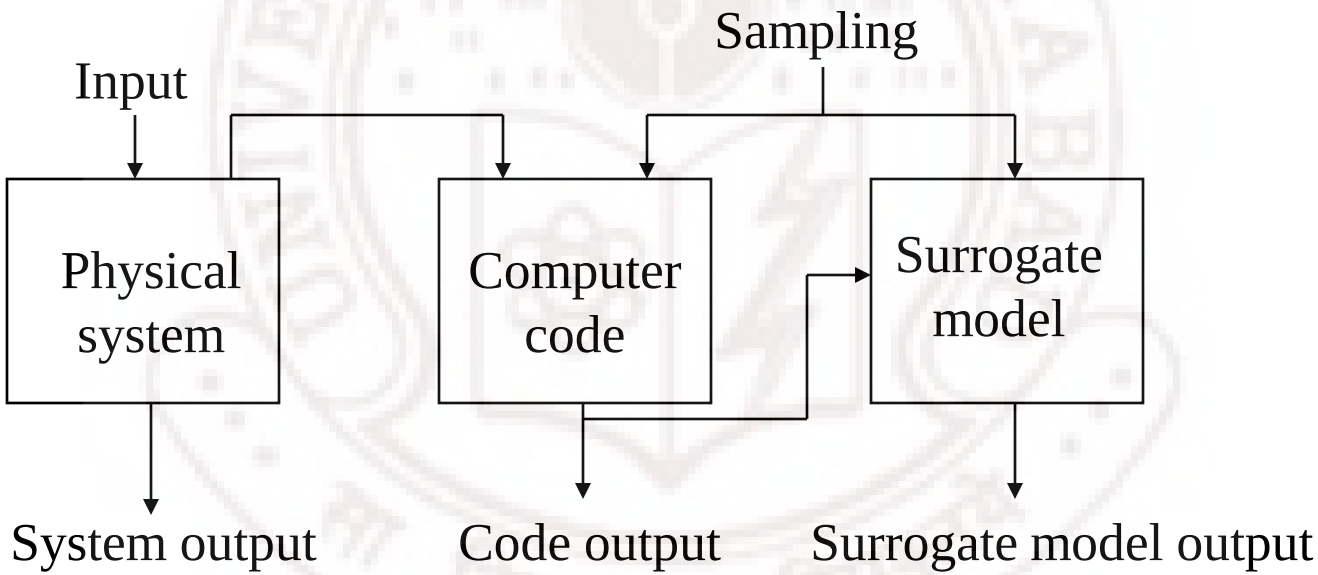


Figure :SURROGATE MODELLING PHILOSOPHY

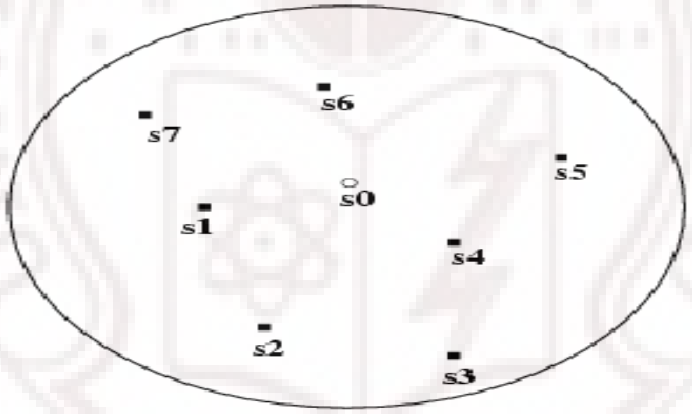
DACE :Design and Analysis of Computer Experiments

- DACE is a Matlab toolbox for working with kriging approximations to computer models [1].
- DACE is used to construct a kriging approximation model based on data from a computer experiment, and to use this approximation model as a surrogate for computer model

[1] Soren N.Lophaven, Hans Bruun Nielsen and Jacob Sondergaard, “DACE A MATLAB Kriging Toolbox”, IMM- Technical University of Denmark, August 1, 2002, version 2.0.

KRIGING APPROXIMATION

- Kriging is a spatial interpolation technique developed by D.G. Krige, a South African mining engineer.
- Below figure shows the values at points s1, s2, . . . are known and value at point s0 is predicted.



- The entire framework of kriging provides a “model of a model” called as Design and Analysis of Computer Experiments

[7] Amitay Isaacs, “Direct-search methods and DACE”, Department of Aerospace Engineering , Indian Institute of Technology Bombay, May 2003.

MODELLING

- The Modeling of deterministic computer response Y(x) is given by

$$\mathbf{Y(x) = f^{T}(x)\,\beta + Z(x)}$$

with: $f(x) = [f_1(x), f_2(x), \ldots\ldots f_k(x)]^T$

$\beta = [\beta_1, \beta_2, \ldots\ldots \beta_k]^T$

- Random Process 'Z' is assumed to have zero mean and constant covariance

$$\text{cov}(x`,x) = \sigma^2 R(x`,x)$$

[3] Etman, L.F.P, "Design and analysis of computer experiments: The method of Sacks et al", Engineering Mechanics report WFW 94.098, Eindhoven University of Technology, 1994.

ESTIMATION

- Given s= { s_1, s_2,…….. s_N } and response data

$$y_s = \{ y(s_1), y(s_2), \ldots\ldots\ldots y(s_N) \}$$

$$\mathbf{y_s = f^T(s)\,\beta + \sigma^2\, R(s,s`)}$$

- From these computer responses the unknown parameters β and σ^2 can be estimated

$$\mathbf{\beta = (F^T R^{-1} F)^{-1} F^T R^{-1} y_s}$$

$$\mathbf{\sigma^2 = 1/N\,(y_s - F\,\beta)^T R^{-1} (y_s - F\,\beta)}$$

with $F = [f(s_1), f(s_2), \ldots\ldots f(s_N)]^T$

$R = [R (s_i , s_j)]_{ij}$ $1 \le i,j \le N$

[3] Etman, L.F.P, “Design and analysis of computer experiments: The method of Sacks et al”, Engineering Mechanics report WFW 94.098, Eindhoven University of Technology, 1994.

PREDICTION

- The Best linear prediction of the response is

$$\hat{y}(x) = f^T(x)\,\beta + r^T(x)\,\gamma$$

where $\gamma = R^{-1}(y_s - F\beta)$

and with correlations 'r'

$r(x) = [R(s_1, x), R(s_2, x), \ldots\ldots R(s_N, x)]^T$

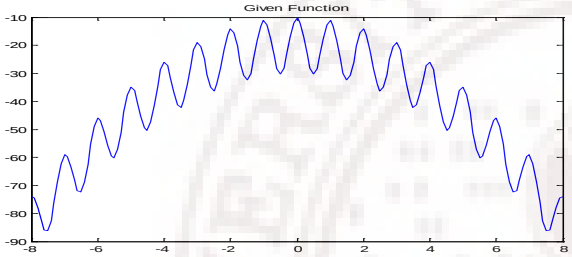
- Calculate mean square error using

$$MSE = E(y(x) - \hat{y}(x))^2$$

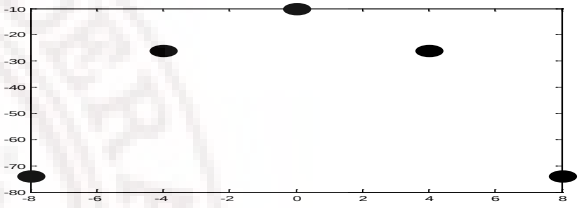
[2] Etman, L.F.P, "Design and analysis of computer experiments: The method of Sacks et al", Engineering Mechanics report WFW 94.098, Eindhoven University of Technology, 1994.

PREDICTION USING DACE MODEL

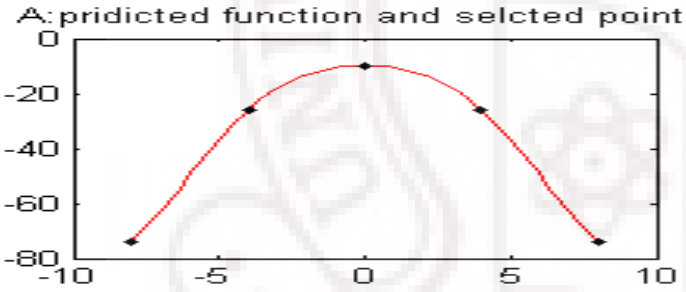
Rastringin's function

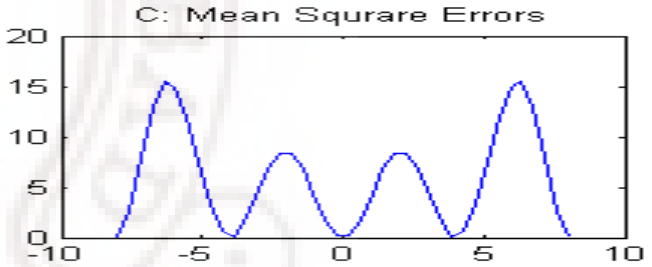


Initial sample points

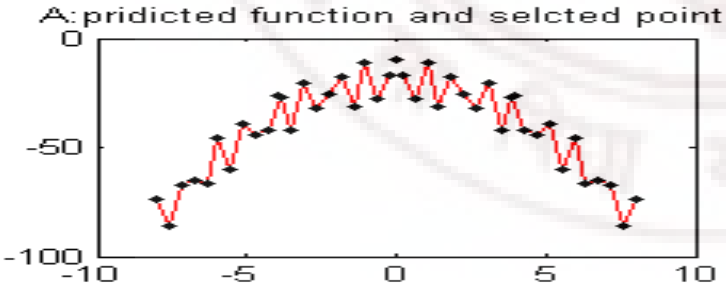


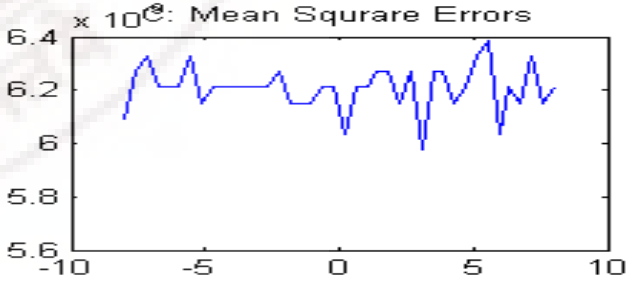
ITERATION:1





ITERATION:15





INTRODUCTION TO OPTIMIZATION

- Optimization comprises a wide variety of techniques from operations research, Artificial intelligence and computer science
- Standard numerical optimization techniques require large amount of evolutions to provide satisfying solutions.
- Use of surrogate models for real process seems beneficial.

[2].D.Biermann,K.Weinert,T.Wagner,"Model-Based Optimization Revisited:Towards Real-Word Processes",IEEE congress on Evolutionary Computation (CEC 2008),PP.2975-2982.

EGO

- Efficient Global Optimization is one of the Model based optimization technique.
- The concept of Efficient Global Optimization (EGO) is developed by Schonlau (1997) and Jones et al. (1998)
- DACE concept was first used for optimization process in EGO

EGO ALGORITHM

1. An initial set of input parameters is selected using sampling.
2. The true objective function is evaluated for all new members of the set.
3. A Kriging surrogate model is fit to the values of the objective function.
4. The expected improvement objective function, calculated using values from the computationally inexpensive Kriging model, is minimized using any suitable global optimization method.[10]

$$\mathbf{E}[I] = (y_{min} - \hat{y}(x))\,\mathrm{CDF}\left(\frac{y_{min} - \hat{y}(x)}{s(x)}\right) + s(x)\,\mathrm{PDF}\left(\frac{y_{min} - \hat{y}(x)}{s(x)}\right).$$

Where s(x) is estimated standard deviation,
CDF is cumulative density function,
PDF probability density function

[10].Rick C.Morgans,Anthony C.Zandder,Colin H.Hansen,David J.Murphy,"EGO Shape Optimization of horn-loaded loudspeakers", Springer science+business media,Nov2007

EGO ALGORITHM(CONTI…)

5. The result of the minimization (the next input parameters most likely to improve the true objective function) is added to the set.
6. The process repeats from step 2 until a predetermined number of iterations is reached.

- The expectation of improvement in our algorithm is given by complementary cumulative distribution function,

$$\text{Probability} = 1 - \text{cdf}(y_{max}, \hat{y})$$

Where ymax is point where maximum response is found and $\hat{y}$ is reponses at trial sites.

Limitations

- The DACE configuration proposed by sacks et al. is not suitable for implementation in a design optimization tool because of unreliable mean squared error[3]
- There is no proof of global convergence for the EGO algorithm[10].
- Towards the end of an optimization run, the conventional EGO algorithm will tend to sample points near previously sampled points[11]

[11]. Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions.Journal of Global Optimum 13:455–492

[2] Etman, L.F.P, “Design and analysis of computer experiments: The method of Sacks et al”, Engineering Mechanics report WFW 94.098, Eindhoven University of Technology, 1994.

[10].Rick C.Morgans,Anthony C.Zandder,Colin H.Hansen,David J.Murphy,”EGO Shape Optimization of horn-loaded loudspeakers", Springer science+business media,Nov2007

Concept of Symmetry

- An object is *symmetric* with respect to a given operation if this operation, when applied to the object, does not appear to change it.
- For a point in two dimensional space (a,b), eight symmetric points can be added.
- EGO algorithm is extended by adding 8 symmetric points for the one best point found.

TEST FUNCTIONS

- Dejong's function $f_1(x) = \sum_{i=1}^{n} x_i^2$
- Rosenbrock's valley : $f_2(x) = \sum_{n-1} 100.(x_{i+1} - x_i^2)^2 + (1 - x_i)^2$
- Rastrigin's function : $f_6(x) = 10.n \sum_{n} (x_i^2 - 10.cos(2.\pi.x_i))$
- Schwefel's function : $f_7(x) = \sum_{i=1}^{n} -x_i.sin(\sqrt{|x_i|})$
- Griewangk's function : $f_8(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
- Branin's function: $f(x_1, x_2) = a(x_2 - bx_1^2 + cx_1 - d)^2 + e(1 - f)\cos(x_1) + e.$
- Goldstein Price's function:

$$f(x_1, x_2) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2]$$
$$[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2].$$

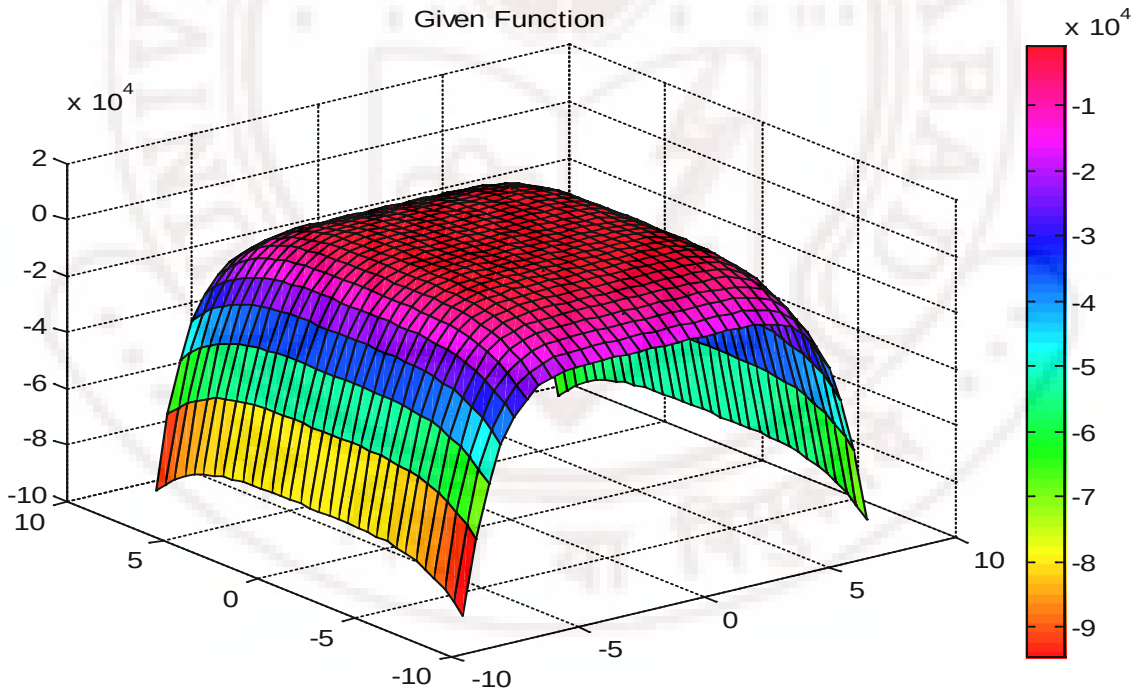
- Six hump camel back function:

$$f(x_1, x_2) = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2.$$

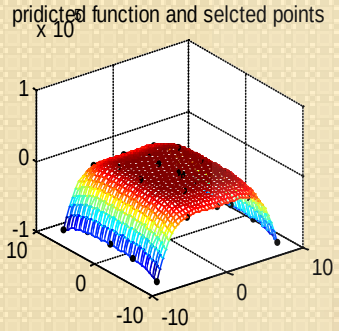
CASE STUDY

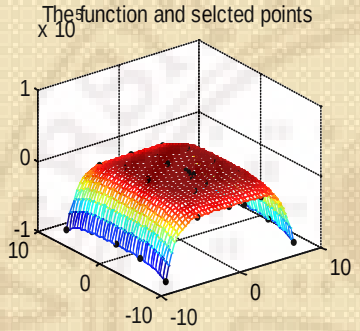
Six hump camel back function $f(x_1, x_2) = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2.$

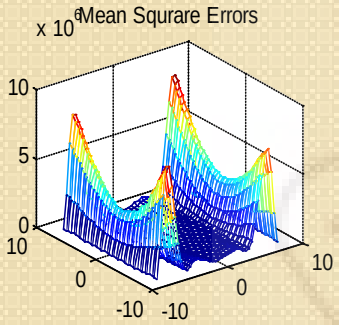
Test area for the function is restricted to the square $-8<x_1<8, -8<x_2<8$. Its Global Maximum is 0.9028 obtainable at [0.2051 -0.6154]

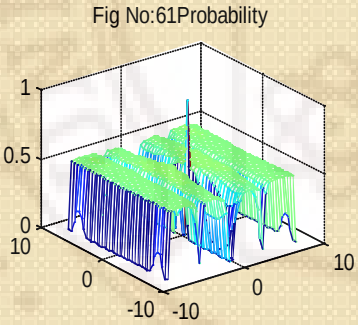


EGO Sample rate-4 ITERATION6

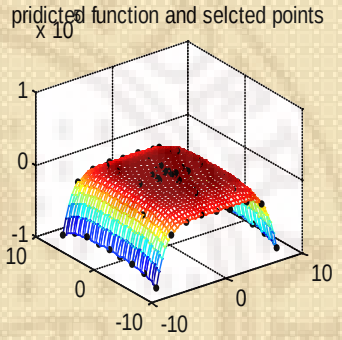


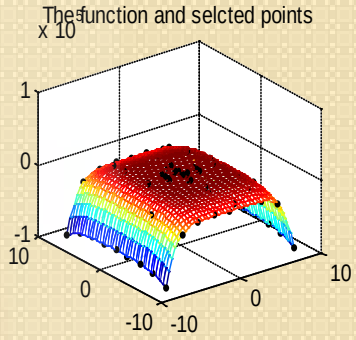


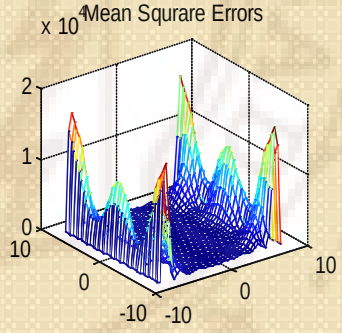


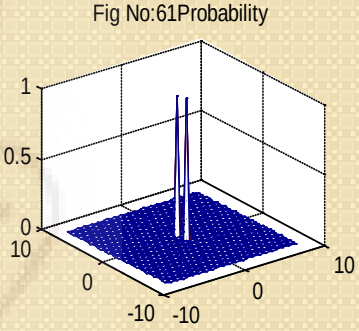


EGO with symmetric points sample 4 iteration 6

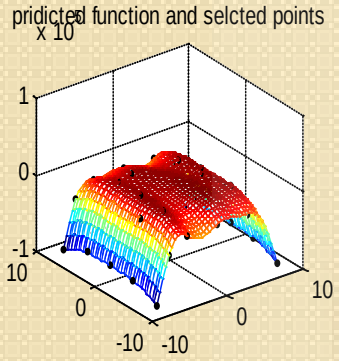


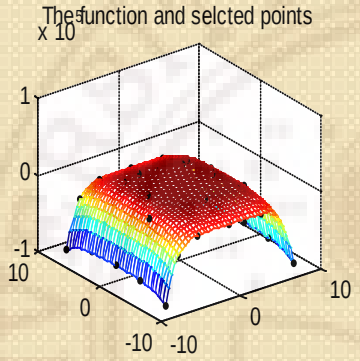


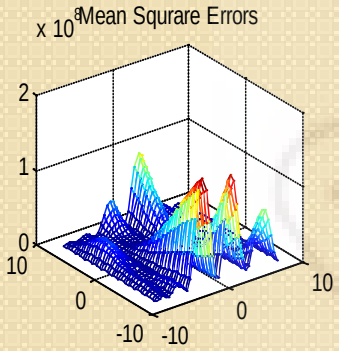


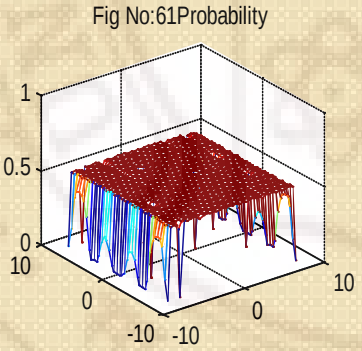


MSE sample rate 4-Iteration 6

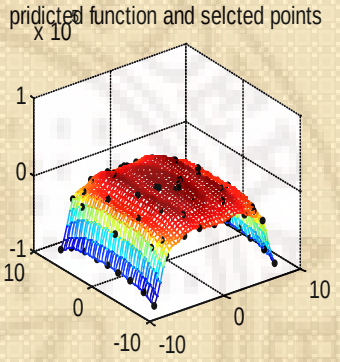


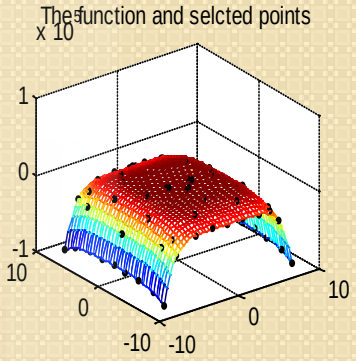


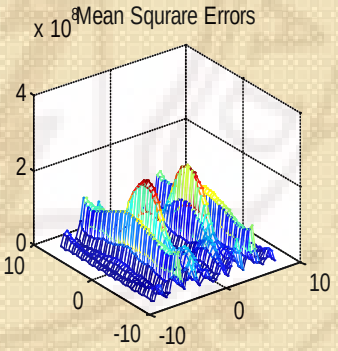


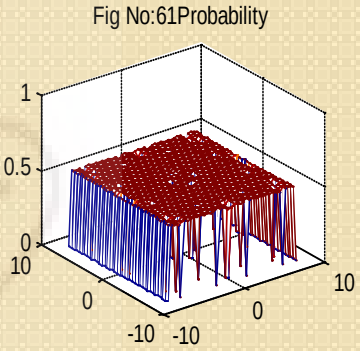


MSE taking symmetric points Iteration 6

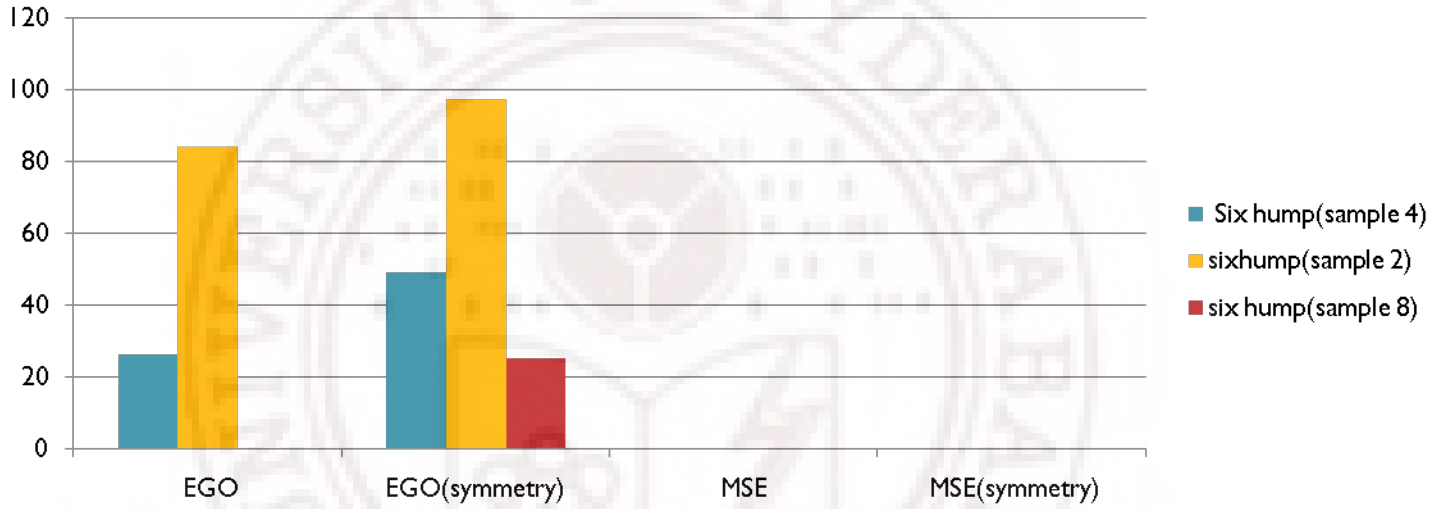








Performance analysis



The function is compared taking different sampling rate and four different cases. While both MSE and MSE with symmetric points fail to converge to global maximum, EGO and EGO with symmetric points took following number of iterations:

	Sample 2:	Sample 4:	Sample8:
EGO	3	6	Local optimum
EGO(symmetric)	2	3	2

SUMMARY-FUTURE WORK

- DACE is chosen as our Surrogate model. Although EGO performs best in providing models for optimization, it would be desirable to obtain stable results under all possible designs and noisy evolutions. The concept of symmetric points has been introduced in EGO which would serve as an enhanced surrogate model to get an unbiased global optimization .Different benchmark test functions have been collected and the efficiency of EGO with symmetric points is being studied.
- How can we take symmetric points for higher dimensional functions?
- Multiobjective optimization using EGO.

REFERENCES

1. Soren N.Lophaven, Hans Bruun Nielsen and Jacob Sondergaard, “DACE A MATLAB Kriging Toolbox”, IMM-Technical University of Denmark, August 1, 2002, version 2.0.
2. D.Biermann,K.Weinert,T.Wagner,“Model-Based Optimization Revisited:Towards Real-Word Processes”,IEEE congress on Evolutionary Computation (CEC 2008),PP.2975-2982.
3. Etman, L.F.P, “Design and analysis of computer experiments: The method of Sacks et al”, Engineering Mechanics report WFW 94.098, Eindhoven University of Technology, 1994.
4. Thomas J. Santner, Brian J. Williams, William I. Notz, “The Design and Analysis of Computer Experiments”, Springer series in statistics, 2003.

REFERENCES

5. Kai-Tai Fang, Runze Li, and Agus Sudjianto, “Design and Modeling for Computer Experiments”, Computer Science and Data Analysis Series, Published by CRC Press, 2006
6. J. Sacks, W.J. Welch, T.J. Mitchell, H.P. Wynn, “Design and Analysis of Computer Experiments”, Statistical Science, vol. 4, no. 4, pp. 409-435, 1989.
7. Amitay Isaacs, “Direct-search methods and DACE”, Department of Aerospace Engineering , Indian Institute of Technology Bombay, May 2003.
8. P.N.Koch,J.P.Evans and D.Powell, ”Interdigitation for effective design space exploration using iSIGHT”, struct Multidisc Optim 23, 111-126 @springer-verlag 2002.

REFERENCES

9. Ankur kulkarni,Prof.P.Mujumdar,”Adaptive Sampling based sampling strategies for DACE surrogate model for expensive black-box functions, ”Department of aerospace engineering, Indian Institute Of Technology Bombay, April 2006.
10. Rick C.Morgans,Anthony C.Zandder,Colin H.Hansen,David J.Murphy,”EGO Shape Optimization of horn-loaded loudspeakers", Springer science+business media,Nov2007.
11. Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions.Journal of Global Optimum 13:455–492.

THANK YOU