The Pennsylvania State University

The Graduate School

Eberly College of Science

A FULLY BAYESIAN APPROACH TO THE EFFICIENT GLOBAL OPTIMIZATION ALGORITHM

A Thesis in
Statistics
by
Sam D. Tajbakhsh

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2013

The thesis of Sam D. Tajbakhsh was reviewed and approved* by the following:

James L. Rosenberger
Professor of Statistics
Thesis Adviser

Enrique del Castillo
Distinguished Professor of Industrial Engineering and Professor of Statistics

David Hunter
Professor of Statistics
Head of the Department of Statistics

*Signatures are on file in the Graduate School.

Abstract

Finding the global optimum(s) of a non-convex function is of great importance in numerous applications in science and engineering where the function takes the form of an expensive computer code and its inputs are the independent variables. For this type of problem, Jones et al. [1] proposed the idea of expected improvement (EI) and embedded it in an algorithm called efficient global optimization, or EGO. Neither EI nor EGO consider the uncertainty in the parameter estimates. One way to account for these uncertainties is to use Bootstrapping. In this paper, instead, we formulate the expected improvement method from a fully Bayesian perspective which results in a corresponding Bayesian EGO method. The performance of the proposed Bayesian EGO is illustrated and compared with the classic EGO method of Jones et al. and the bootstrapped EGO of Kleijnen et al. [2]. Furthermore, we apply the Bayesian EGO algorithm for the optimization of a stochastic inventory simulation model. It is shown how a bayesian approach to EGO allows to optimize not only the expected improvement criterion, but also any function of the posterior predictive density, such as quantile, leading to a bayesian expected quantile improvement method.

Contents

List of Figures

List of Tables

Dedication

I would like to dedicate this thesis to my dad who has devoted his life to me since I was born and to my lovely wife for being beside me during the past six years.

Acknowledgments

I would like to take this opportunity to express my deepest gratitude to my advisor, Dr. James Rosenberger. Without his support and believe in me as a student from engineering field, I should have quite the program long time ago. Not only is he a true academician, but also an excellent example of a person to follow in the social life.

Also, I am deeply grateful to have Dr. Enrique del Castillo throughout these four years. He was the reason I chose Penn State to pursue my graduate studies. I appreciate the energy he put in the classes to impart knowledge to his students, the time he spent on revising this thesis, and the advices he gave me in my research. He is a true inspiration for the research excellence.

1 Introduction

In numerous applications in science and engineering, the input/output relation of a system is frequently represented by a computer code with certain inputs we wish to optimize. For instance, finite element models in aerospace mechanical design and systems of partial differential equations (PDE) that model chemical reactions in a petrochemical plant or fluid dynamical systems as used in engineering and geography. The computer codes can be regarded as a function mapping from the input space to the output (response) space. Functions of this type usually share two properties: 1) they are computationally expensive, that is, each run of the code takes considerable amount of time and 2) they are highly non-convex. These two properties make the global optimization of these functions very challenging.

Global optimization has a large body of literature which has traditionally focused on deterministic functions. There are many books, monographs and papers in which researchers have developed a variety of approaches to find the global optimum(s) of nonconvex functions. A complete review of this literature is outside the scope of the present thesis. Instead, a thorough review of global optimization methods based on statistical models is given in Chapter 2.

1.1 Metamodeling

A *metamodel* is a statistical model used to approximate an expensive computer code. As discussed in [3], a major branch of global optimization is based on building easy

to evaluate statistical models that allow one to interpolate the function of interest and then optimize it. Polynomial regression, splines, kriging, radial basis functions, neural networks and support vector machines (SVM) have been used as metamodels in different applications. For a review of these techniques, readers may refer to [4].

Among these statistical models, kriging has probably received the most attention. The term "kriging" which is borrowed from the name of a mining engineer, Danie Krige, was initially a distance-weighted average interpolator for gold level in the mines [5]. The mathematical theory of Kriging, and the term kriging itself is due to the French mathematician Georges Matheron [6] and is still an ongoing research interest in geostatistics. The important property which makes kriging an attractive statistical technique for approximating computer codes is that it is an *exact interpolator*. In other words, in the absence of random error, the kriging predictions perfectly match the observed data [7], [8]. Below, we briefly review prediction based on kriging.

1.1.1 Kriging

Consider a setup in which the function of interest is observed at a set of locations $X = (\mathbf{x_1}, ..., \mathbf{x_n})^T$ where $\mathbf{x_i} \in D \subset \mathbb{R}^d$ and the vector of observed function values (responses) is $\mathbf{y} = (y_1, ..., y_n)^T$. We are interested in predicting the function value y_0 at the location x_0. In a statistical modeling approach, we assume that the observed data are realizations of a random field; hence, y_0 is a random variable that needs to be predicted by $\widehat{y}_0(\mathbf{y})$.

Note - The term random *field* is mostly used when $\mathbf{x} \in D \subset \mathbb{R}^d$ and $d > 1$; however, when $d = 1$ random *process* is more dominant. In this thesis, these two terms have been used interchangeably.

The kriging predictor is a member of the class of *linear unbiased* predictors. It

is linear since it is a linear function of the observed data of the form $\widehat{y}_0 = a_0 + \mathbf{a}^T\mathbf{y}$ and it is unbiased with respect to the class of distributions $\mathcal{F}$ since $E_F\{\widehat{y}_0\} = E_F\{y_0\}$ where $E_F\{.\}$ denotes expectation under $F(.)$, the distribution of $(y_0, \mathbf{y})$, for all $F \in \mathcal{F}$ [8].

The kriging predictor results in a minimum *mean square prediction error* (MSPE). The MSPE of $\widehat{y}_0(\mathbf{y})$ is

$$\text{MSPE}(\widehat{y}_0, F) \equiv E_F\{(\widehat{y}_0 - y_0)^2\} \tag{1}$$

A central theorem of prediction indicates that the conditional mean of y_0 given $\mathbf{y}$ is indeed the minimum MSPE predictor (see [8], Theorem 3.2.1), i.e. $\widehat{y}_0 = E\{y_0|\mathbf{y}\}$ is the minimum MSPE linear unbiased predictor. The minimum MSPE linear unbiased predictor is also known as *best linear unbiased predictor* or BLUP [8].

In a great portion of the literature on kriging prediction, the underlying random process is assumed to be stationary Gaussian Process (GP) which results in the following model:

$$y(\mathbf{x}) = \mu(\mathbf{x}) + z(\mathbf{x}) \tag{2}$$

where $\mu(\mathbf{x})$ is the mean of the stochastic process which is usually assumed to have the form $\mathbf{f}^T(\mathbf{x})\boldsymbol{\beta}$, $z(\mathbf{x})$ is a zero mean stationary Gaussian process with

$$\text{Cov}\{z(\mathbf{x}_i), z(\mathbf{x}_j)\} = \sigma^2 R(\mathbf{x}_i - \mathbf{x}_j) \tag{3}$$

for some correlation function $R(.)$. The model above provides the joint distribution of $y_0 = y(\mathbf{x}_0)$ and $\mathbf{y} = (y_1, ..., y_n)^T$ as

$$\begin{pmatrix} y_0 \\ \mathbf{y} \end{pmatrix} \sim N_{1+n}\left[\begin{pmatrix} \mathbf{f_0}^T \\ F \end{pmatrix}\boldsymbol{\beta}, \sigma^2 \begin{pmatrix} 1 & \mathbf{r_0}^T \\ \mathbf{r_0} & R \end{pmatrix}\right] \tag{4}$$

where $\mathbf{f_0} = \mathbf{f}(\mathbf{x_0})$ is the $p \times 1$ vector of regressors at $\mathbf{x_0}$, F is $n \times p$ matrix of regressors at n observations, $\boldsymbol{\beta}$ is $p \times 1$ vector of unknown mean parameters, $\mathbf{r_0}$ is $n \times 1$ vector of correlations of the new point with the n observed points $\mathbf{r_0} = (R(\mathbf{x_0} - \mathbf{x_1}), ..., R(\mathbf{x_0} - \mathbf{x_n}))^T$ where R is $n \times n$ matrix of correlations between the observations $R = [R(\mathbf{x_i} - \mathbf{x_j})], (i, j) \in \{1, ..., n\}$. Assuming that the design matrix F is a full column rank matrix and R is a positive definite matrix, it can be shown [8] that the minimum MSPE linear unbiased predictor $\widehat{y}(\mathbf{x_0}) = E\{y(\mathbf{x_0})|\mathbf{y}\}$ is

$$\widehat{y}(\mathbf{x_0}) = \mathbf{f_0}^T\boldsymbol{\beta} + \mathbf{r_0}^T R^{-1}(\mathbf{y} - F\boldsymbol{\beta}) \tag{5}$$

Assuming that R is known, $\boldsymbol{\beta}$ can be replaced by its least squares estimate $\widehat{\boldsymbol{\beta}} = (F^T R^{-1} F)^{-1} F^T R^{-1}\mathbf{y}$, the resulting formula is known as the universal kriging. Universal kriging was proposed by Sacks et al. [9] as a metamodel to approximate a computer code. Notice that the class of distributions $\mathcal{F}$ for which the above predictor is minimum MSPE is very small [8].

It can be shown that the MSPE of the above predictor is

$$s^2_{\widehat{y}}(\mathbf{x_0}) = \sigma^2 \left\{1 - \mathbf{r_0}^T R^{-1}\mathbf{r_0} + \mathbf{h}^T (F^T R^{-1} F)^{-1}\mathbf{h}\right\} \tag{6}$$

where $\mathbf{h} = \mathbf{f_0} - F^T R^{-1}\mathbf{r_0}$. Here, σ^2 is usually replaced by its maximum likelihood estimate $\widehat{\sigma^2} = \frac{1}{n}\left(\mathbf{y} - F\widehat{\boldsymbol{\beta}}\right)^T R^{-1}\left(\mathbf{y} - F\widehat{\boldsymbol{\beta}}\right)$. We would like to highlight two points: first, the MSPE is equal to zero for observed data which means that $\widehat{y}$ is an *exact interpolator* $\widehat{y}(\mathbf{x}_i) = y(\mathbf{x}_i)$, $i \in \{1, ..., n\}$; second, the MSPE formula is independent of $\mathbf{y}$ which means that the observations are homoscedastic.

Note - It is worthy to mention that the derivation of the formulas (5) and (6) is also achieved by directly minimizing MSPE (1) for the linear predictor; hence, the assumption of normality only contributes to the estimation of unknown parameters of the model.

1.1.2 Stochastic Models for Global Optimization

An up-to-date review of stochastic processes used for global optimization is provided by Zhigljavsky and Zilinskas [3] who discuss the Wiener process for the approximation of one-dimensional functions in detail. The Wiener process $\xi(x), x \geq 0$ is a Gaussian process with zero mean and covariance function $\mathrm{Cov}(\xi(x_1), \xi(x_2)) = \sigma^2 \min(x_1, x_2)$, where σ^2 is a parameter; the increments of the Wiener process $\xi(x+\delta) - \xi(x)$ are $N(0, \delta\sigma^2)$ random variables and the increments corresponding to disjoint time intervals are independent. Hence, if the target function is highly multimodal with many local optima then the property of independence of increments for disjoint intervals is reasonable. Since the Wiener process is Markovian, a Wiener process assumption is also favorable from a computational point of view. However, since the sampling functions (realizations) of the Wiener process are not differentiable almost everywhere with probability one, its local properties are under criticism in application.

Zhigljavsky and Zilinskas [3] also review stationary Gaussian processes as the general class of statistical models which are defined by the mean, variance and correlation function. The desired smoothness of their realizations can be controlled by the behavior of the correlation function in the neighborhood of zero. The Ornstein-Uhlenbeck process is the only Markov process in the class of stationary Gaussian processes. Its correlation function is $\exp(-c|x|), c > 0$ which is not differentiable at $t = 0$ and hence its realizations are not differentiable. Therefore, to enforce the smooth realization property we must give up the Markov property. However, the implementation difficulties in the non-Markovian processes kept global optimization researchers away from using smooth stationary stochastic processes. We should note that not all properties of the random processes can be generalized to random fields, stochastic functions of many variables. For instance, the gaussian homogenous isotropic random field with exponential correlation function is not Markovian.

1.2 Research Objectives

This dissertation deals with finding the global optimum of non-convex, expensive to evaluate and black box type objective functions based on statistical modeling. The objective function values are assumed to be realizations of a stationary Gaussian process and the Kriging predictor is used. The approach is categorized as one-stage ahead bayesian optimization based on the Expected Improvement (EI) criterion. By one-stage ahead optimization we mean that we want to optimize the function given the current information (the function values at some points) at the very next step and not over a finite horizon [10]. Finally, the approach is *fully* bayesian, i.e. all of the parameters of the model are assumed to be random variables with noninformative priors.

1.3 Dissertation Outline

In this dissertation, we propose a fully bayesian approach to evaluate the expected improvement criterion at any point $\mathbf{x}$ based on the posterior predictive distribution. This is then embedded in an optimization method which we call "bayesian EGO". The rest of the dissertation is as follows.

In Chapter 2, I review the literature of the bayesian optimization; specifically, the classic expected improvement and the Efficient Global Optimization (EGO) algorithm and related methods are described. I also address the problem created by using plug-in parameter estimates in the Kriging prediction variance which is the major motivation behind this thesis. Chapter 3 contains our proposed bayesian EGO approach and also discusses some details on computation of the bayesian expected improvement. In chapter 4, our proposed model is first validated through some simulation experiments; then, the performance of the classic EGO, bootrapped EGO and bayesian EGO are compared through several deterministic test functions, and finally,

the proposed approach is implemented for optimization of a stochastic simulation computer code from the inventory control literature. Chapter 5 gives concluding remarks and directions for future research.

2 Literature Review

Global optimization is a growing area which has many applications in engineering, computational chemistry, finance, medicine and many other fields. Although classical optimization theory cannot directly be applied to global optimization problems, many convex optimization tools are being used in global optimization methods (e.g. "branch and bound" type methods). In general, theory of deterministic global optimization is much better developed compared to stochastic global optimization. However, for *black box* optimization where information about the deterministic model is mostly the objective function value at a certain input and non-availability of derivatives prevents one from using gradient-based techniques, stochastic approaches are shown to be more promising. The stochastic approaches are also useful when dealing with stochastic function where the objective function is corrupted by noise.

Some of these algorithms are based on analogies with natural processes e.g. Genetic Algorithms (GA) or Simulated Annealing (SA). The efficiency of these search methods, however, strongly depends on fast evaluation of the objective function. Therefore, for expensive to evaluate computer codes where one run of the function takes considerable amount of time, they are not suitable. Some other approaches are based on stochastic modeling of the underlying function which result into a cheap to evaluate statistical approximation which can then be optimized. Reviewing these approaches is major focus of this chapter.

2.1 Bayesian Optimization

The earliest work found which uses stochastic modeling is by Kushner [11]. He considered a one-dimensional function $y(x)$ which could be multi-modal and may not be continuous or differentiable. Letting $y(x) = z(x) + \eta(x)$ where $z(x)$ is the process state and $\eta(x) \sim N(0, \sigma_x^2)$ is random noise, he proposed to model the state with a Wiener process, i.e.,

$$z(x) = z(x') + \xi(x, x') \tag{7}$$

where $\xi(x, x') \sim N(0, c|x - x'|)$. The observation noise are assumed to be independent and independent of $z(x)$. For prediction purposes, the expectation and variance of $z(x)$ conditioned on observations $y(x_i), i = 1, .., n$ is desired. However, since $z(x)$ is a process of independent increments, it is only well defined in term of differences. Hence, the state and function value differences were defined as $\Delta z(x) = z(x) - z(0)$ and $\Delta y_i = y_i - z(0)$ where $z(0)$ is an arbitrary observation at present and the conditional expectation and variance of $\Delta z(x)$ given $\Delta y_i, i = 1, ..., n$ were calculated. Notice that the $\Delta z(x)$ and Δy_i has a joint normal distribution. The conditional expectations involve calculation of at least $n + 1$ cofactors of the covariance matrix. Furthermore, $z(0)$ was set to 0; hence, $\Delta z(x) = z(x)$ and $\Delta y_i = y_i$. This restriction was then removed by taking the limit as $x \to -\infty$. This indirectly helps in calculating the process, $z(x)$, conditional expectation and variance rather than its difference while it still requires evaluation of at least n determinant of order n in addition to much other calculations. Hence, the proposed approach which is for one-dimensional curve fitting is computationally expensive.

Mockus et al. [12] proposed the idea of bayesian optimization as minimization of the expected deviation from the extremum to find the global minimum of the function. The expected deviation which is the function of the policy π (the method

for selecting the solutions) and the time horizon N is defined as

$$\delta(\pi, N) = E\{\xi(x_N^*) - \min_{x \in A} \xi(x)\} \tag{8}$$

where x_N^* denotes the minimizer of the stochastic process obtained from the policy π after N steps. The optimal policy $\pi^*(N)$ which minimizes the above expected deviation with π as the argument is the solution to the following recurrent equations:

$$\begin{aligned}
u_k\left[(x_i, y_i), i = 1, ..., k-1\right] &= \min_{x \in A} E\{u_{k+1}\left[(x_i, y_i), (x, \xi(x))\right] | \xi(x_i) = y_i, i = 1, ..., k-1\} \\
\pi_k^*\left[(x_i, y_i), i = 1, ..., k-1\right] &= \arg\min_{x \in A} E\{u_{k+1}\left[(x_i, y_i), (x, \xi(x))\right] | \xi(x_i) = y_i, i = 1, ..., k-1\} \\
k &= N, N-1, ..., 2 \\
u_1 &= \min_{x \in A} E\{u_2\left[(x, \xi(x))\right]\} \\
\pi_1^* &= \min_{x \in A} E\{u_2\left[(x, \xi(x))\right]\}
\end{aligned} \tag{9}$$

where $\xi(x)$ is a realization of some stochastic process. However, in [12] a one-stage bayesian method ($N = 2$) was proposed as a simplified version of the optimal method. To select the corresponding stochastic process, they mentioned some conditions that the stochastic process should satisfy to be eligible such as continuity of realization and independence of the n^{th} difference (as discrete approximation of n^{th} derivative). Furthermore, the selected stochastic process should be homogenous on its domain. Gaussian process and Wiener process as two stochastic processes which satisfy the aforementioned conditions were proposed and the parameter estimates for some cases were provided [12]. Zilinskas [13] also provided some algorithmic implementation details for optimization of one-dimensional multimodal function by the one-stage bayesian method based on the Wiener process. In a different paper, Zilinskas [14] argued the effectiveness of a Wiener process for smooth functions and the fact that

the sampling function of a Wiener process is not differentiable almost anywhere and took an axiomatic approach on selection of a descent stochastic process.

A version of one-dimensional one-step bayesian algorithm based on Wiener process was proposed in [15]. To determine the next observation the following equation was proposed:

$$x_{n+1} = \arg\min_{0 \leq x \leq 1} E\{\min(\xi(x), y_{min}) | \xi(x_1) = y_1, ..., \xi(x_n) = y_n\} \tag{10}$$

where $\xi(x)$ is the underlying stochastic model and y_{min} is the minimum of the observed function values. This approach performs a very local search concentrating the observations in the vicinity of the best point found. The local behavior of the approach motivated Zilinskas to search for a more global-oriented optimization method. He investigated the paper by Kushner [11] and provided some theoretical justifications for the approach and proposed the P-algorithm [16] for the multidimensional case using random fields ($\dim(\mathbf{x}) > 1$). The P-algorithm suggests the location for the next function evaluation as

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x} \in A} P\{\xi(\mathbf{x}) \leq \widetilde{y}_{min} | \xi(\mathbf{x}_1) = y_1, ..., \xi(\mathbf{x}_n) = y_n\} \tag{11}$$

where $\widetilde{y}_{min} = y_{min} - \epsilon_n$ and ϵ_n is a parameter. The properties of the P-algorithm depends on the chosen stochastic process and the parameter ϵ_n. For one-dimensional case the Wiener process is proposed as the statistical model. However, for higher dimensions, a random field with the favorable Markov property is not known. Hence, a generalized version of the Wiener process was proposed for two and three dimensional cases based on a procedure called *cloning*. In the cloning procedure, the input space is repeatedly divided into equilateral triangles and the observations are only allowed on the vertices of these triangles. The Markov property is then enforced by conditioning on the observations on the neighboring vertices. For more details refer to [16].

The globality vs. locality characteristic of the P-algorithm approach can be con-

trolled by the parameter ϵ_n. For small ϵ_n the approach tends to be more local while as $\epsilon_n \to \infty$ the P-algorithm suggests points of maximal uncertainty (for more details see [17], [3]). In the context of computer experiments and optimization through metamodeling, this is known as a tradeoff between *exploration* and *exploitation*. By exploration, we mean searching the experimental domain and escaping from local optima, and by exploitation we mean moving toward the global optimum as close as possible. In the control theory literature, this is known as "dual control" problem. Most of the procedures which use metamodels consist of sequential iterations between parameter estimation (rebuilding the model) and optimization of the model at the given iteration. These methods are mainly discussed in the context of sequential design of optimization experiments ([18], [19], [8] and [20]).

2.2 Expected Improvement (EI) and the Efficient Global Optimization (EGO) Algorithm

In a seminal paper in bayesian optimization, Jones et al. [1] proposed the notion of Expected Improvement (EI) based on the work by Mockus et al. [12]. Their method combines the mean and variance structures of the kriging predictor such that it explores the experimental domain and at the same time exploits the potential areas where local optima occur. Jones et al. [1] named the resulting algorithm *Efficient Global Optimization* (EGO)– see also [21] and [22]. As defined in [1], the Expected Improvement (EI) at point $\mathbf{x}$ is

$$E[I(\mathbf{x})] = E[\max(y_{min} - y(\mathbf{x}), 0)] \tag{12}$$

where y_{min} is the current best (minimum) function value. Note that $I(\mathbf{x}) = \max(y_{min} - y(\mathbf{x}), 0)$ is the improvement at the point $\mathbf{x}$ which is a random variable since it is a function of $y(\mathbf{x})$. If $\widehat{y}(\mathbf{x})$ and $s^2(\mathbf{x})$ are the mean and the variance predictors of y at

point $\mathbf{x}$ as in (5) and (6), then assuming $y(\mathbf{x}) \sim N(\widehat{y}(\mathbf{x}), s^2(\mathbf{x}))$, as shown in Appendix A, the expectation in (12) can be simplified to

$$E[I(\mathbf{x})] = (y_{min} - \widehat{y}(\mathbf{x}))\Phi\left(\frac{y_{min} - \widehat{y}(\mathbf{x})}{s(\mathbf{x})}\right) + s(\mathbf{x})\phi\left(\frac{y_{min} - \widehat{y}(\mathbf{x})}{s(\mathbf{x})}\right) \tag{13}$$

where Φ and ϕ are the cdf and the pdf of the standard normal distribution, respectively. The EGO algorithm consists of maximizing the expected improvement criterion given the vector of observed function values $\mathbf{y}$ and the matrix of locations X and obtaining an optimal solution location $\mathbf{x}^*$. We then evaluate the function (computer code) at $\mathbf{x}^*$, find $y(\mathbf{x}^*)$ and add $\mathbf{x}^*$ to the bottom of X and y^* to the bottom of $\mathbf{y}$. This procedure is repeated until a stopping criterion is satisfied.

The notion of expected improvement has gained considerable attention in recent years. Williams et al. [18] used the EI approach for a robust parameter design scenario where they wanted to find the optimal setting of some control variables in the presence of environmental noise variables. The objective function to be minimized is the expected loss where the expectation is taken over the distribution of environmental variables. Ginsbourger and Riche [10] showed the suboptimality of the one-stage ahead maximization of EI at each iteration of the EGO algorithm by means of a counterexample. They further proposed a finite time horizon dynamic programming approach to maximize a multi-stage expected improvement criterion given a finite budget of experimentation. Huang et al. [23] proposed an *augmented* expected improvement criterion for optimizing stochastic responses which accounts for the uncertainty in the current best solution. Recently, Roustant et al. [24] introduced two R packages for kriging-based metamodeling of computer experiments (DiceKriging) and then for its optimization using the expected improvement criterion (DiceOptim).

In a recent paper by Pichney et al. [25], an extension of the expected improvement criterion based on quantiles was proposed for the optimization of *stochastic*

simulators (functions). They assumed that the output of the computer code is observed with noise $\widetilde{y}(\mathbf{x}_i) = y(\mathbf{x}_i) + \epsilon_i$ where the noise $\epsilon_i \sim N(0, \tau_i^2)$ is assumed to be independent and normally distributed random variables and the noise variance τ_i^2 is a monotonically decreasing function of the computer code's computation time t_i. Since the observations are noisy the exact function values are unknown; hence, the approach proposes to use the β-quantiles at the location $\mathbf{x}$ given by the kriging conditional distribution, for a give level $\beta \in [0.5, 1]$ which is denoted by $q(\mathbf{x})$. Then, they defined an improvement I to be the decrease of the lowest β-quantile between the present iteration n and the coming iteration $n+1$:

$$I = (\min(q_n(X_n)) - q_{n+1}(\mathbf{x}_{n+1}))^+ \tag{14}$$

where $(.)^+$ is the positive part operator. Hence, the Expected Quantile Improvement (EQI) is

$$EQI_n(\mathbf{x}^n, \tau_{n+1}^2) = E\left[\left(\min_{i \leq n}(q_n(\mathbf{x}_i)) - q_{n+1}(\mathbf{x}_{n+1})\right)^+ | \widetilde{y}_i, i \in \{1, ..., n\}\right] \tag{15}$$

In the absence of noise ($\tau^2 = 0$), the EQI is equal to the classic EI while a very noisy future observation (big τ^2) can have a limited influence on the kriging model and the improvement function will be mostly zero (see [25], Figure 2). From the other way, the β parameter tunes the exploration/exploitation property of the optimization routine. With $\beta = 0.5$ high prediction variance has significant effect in increasing the EQI; hence, the routine will behave exploratory while as $\beta \longrightarrow 1$ the criterion penalizes points with high prediction variance and the routine favors exploitation ([25], Figure 2).

2.3 The Problem of Using Plug-in Estimates in the Kriging Prediction Variance

Den Hertog et al. [26] and Sjostedt de Luna and Young [27] showed that the formula for estimating the kriging prediction variance equation, (6), underestimates the true prediction variance in expectation when the plug-in parameter estimates (the variance parameter σ^2 and the covariance range parameter ϕ) are inserted in the formula (this issue was also briefly mentioned by Jones et al. [1], p.463). Both papers proposed bootstrapping as a means to estimate the true kriging variance. Den Hertog et al. [26] first calculated the maximum likelihood estimates of the parameters of the kriging model (the mean, variance and correlation parameters) using the original data. The resulting model with MLE parameters is then used to sample the bootstrapped observations.

This idea was adopted by Kleijnen et al. [2] in their bootstrapped EGO algorithm. They incorporated the *parametric* bootstrapped estimate of kriging variance in (13) and named it *bootstrapped* EI. From the original dataset $(X, \mathbf{y})$, they first find the Maximum Likelihood Estimates (MLE) of the kriging parameters and after replacing the parameters with their ML estimates and assuming a Gaussian distribution, they sampled at the new point where the prediction is desired $\mathbf{x}_{new}$, namely $y^*_{new;b}$. They also sampled a bootstrapped dataset at the locations of the original data points X, namely $\mathbf{y}^*_b$. Next, using the bootstrapped dataset, $(X, \mathbf{y}^*_b)$, they calculated the bootstrapped ML estimates of the kriging model and used that model to predict at the new point $\mathbf{x}_{new}$, namely $\hat{y}^*_{new;b}$. This procedure is repeated B times, $b = 1, ..., B$ (where B is the bootstrap sample size), and the bootstrapped variance estimate of the kriging model is estimated as

$$s^2_B(\mathbf{x}_{new}) = \frac{1}{B}\sum_{b=1}^{B}(\hat{y}^*_{new;b} - y^*_{new;b})^2 \tag{16}$$

The *Bootstrapped* EI uses $s_B^2(\mathbf{x}_{new})$ as the true estimate of the kriging prediction variance at any new point. Note that in principle, the whole procedure should be followed for each candidate point $\mathbf{x}_{new}$. However, to speed-up the computations, they use the same bootstrapped MLE computed from $(X, \mathbf{y}_b^*)$ for all candidates.

A major issue to tackle using either classic EI or bootstrapped EI is finding the maximum likelihood estimate of the kriging parameters. Note that using bootstrapped EI, in each iteration of the bootstrapped EGO algorithm, $B+1$ maximization problems need to be solved compared to only one maximization for classic EGO. Generally, the constrained maximization of the likelihood function even for the simplest model structure is not an easy task and the routine may converge to a local maximum.

Benassi et al. [28] also proposed a bayesian method for EGO algorithm. Bayesian formulations of the kriging model have existed for about two decades, see [29]. In [28], the authors assumed that the covariance range parameter (ϕ) is independent of the mean (μ) and variance (σ^2) parameters; furthermore, assuming specific priors for the mean and variance parameters, they derived a closed form formula for expected improvement. However, to account for range parameter uncertainty, the paper proposes approximation techniques.

The proposed approach discussed in the next chapter assumes different prior distribution settings than in [28] and does not assume independence of the range parameter from the other parameters of the model. We propose calculation of the expected improvement based on the posterior predictive distribution at the new location.

3 Proposed Bayesian Expected Improvement Method

Given that the estimated kriging variance with plug-in parameter estimates in the classic EGO is biased and the repeated optimization problems in the bootstrapped EGO are difficult, we take instead a bayesian approach to calculate the expected improvement while considering the uncertainty of the parameters in the predictions. The posterior predictive distribution of $y(x)|\mathbf{y}$ is then used to calculate the expected improvement through its original definition given in (12). The method has the additional advantage over the bootstrapped EGO version that it allows the use of alternative loss functions, such as EQI, since the posterior predictive density of $y(x)|\mathbf{y}$ is calculated.

The model in equation (2) is assumed with a fixed mean and added measurement noise:

$$y(\mathbf{x}) = \mu + z(\mathbf{x}) + \epsilon(\mathbf{x}) \tag{17}$$

where $\epsilon(\mathbf{x})$ are independent normal random variables $\epsilon(\mathbf{x}) \sim N(0, \tau^2)$; also, $\epsilon(\mathbf{x})$ is independent of $z(\mathbf{x})$. Furthermore, we assume that the stochastic process $z(\mathbf{x})$ is isotropic, that is, the covariance between $z(\mathbf{x}_i)$ and $z(\mathbf{x}_j)$ depends only upon the distance between the points $d(\mathbf{x}_i, \mathbf{x}_j)$ and not on the direction of the vector joining them [7]. We use an exponential covariance function to model $\mathrm{Cov}(z(\mathbf{x}_i), z(\mathbf{x}_j))$ which

is popular in the metamodeling literature. This makes $\Sigma_{ij} = \text{Cov}(y(\mathbf{x}_i), y(\mathbf{x}_j)))$ equal to

$$\Sigma_{ij} = \begin{cases} \sigma^2 \exp\left(-\phi d(\mathbf{x}_i, \mathbf{x}_j)\right) & \text{if } d(\mathbf{x}_i, \mathbf{x}_j) > 0 \\ \sigma^2 + \tau^2 & \text{otherwise} \end{cases} \tag{18}$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ is the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. Given the above model, the kriging parameter vector is $\Theta = \{\mu, \phi, \sigma^2, \tau^2\}$ where $\phi, \sigma^2, \tau^2 > 0$. Bayesian implementation of this simple model with only four parameters has shown satisfactory results in approximating functions of different complexity and dimension as is discussed in chapter 4. For the Gaussian process model above, the likelihood is

$$L(\Theta) = (2\pi)^{-n/2} |\Sigma|^{-1/2} \exp\left(\frac{-1}{2}(\mathbf{y} - \mu\mathbf{1})^T \Sigma^{-1} (\mathbf{y} - \mu\mathbf{1})\right) \tag{19}$$

where $\mathbf{y}$ is the vector of function values for an initial design X and $\mathbf{1}$ is $n \times 1$ vector of ones.

Given a set of prior distributions $\pi(\cdot)$ for the parameters of the model, the posterior distribution $p(\Theta|\mathbf{y})$ which is proportional to the likelihood multiplied by the priors, can be derived. We then need to find the posterior predictive distribution of $y(\mathbf{x})|\mathbf{y}$. Using the multivariate normal distribution shown in equation (4) we get

$$\begin{bmatrix} y(\mathbf{x}) \\ \mathbf{y} \end{bmatrix} \sim N_{1+n}\left(\mu\mathbf{1}, \begin{bmatrix} \sigma^2 + \tau^2 & \boldsymbol{\gamma}^T \\ \boldsymbol{\gamma} & \Sigma \end{bmatrix}\right)$$

where $\mathbf{1}$ is now an $(n+1) \times 1$ vector of ones and $\boldsymbol{\gamma}$ is an $n \times 1$ vector of covariances of the new point $\mathbf{x}$ with the observed points X. Given a well-known property of the multivariate Gaussian distribution [30], we obtain the posterior predictive density:

$$p(y(\mathbf{x})|\mathbf{y}, \Theta) = N\left(\mu + \gamma^T \Sigma^{-1}(\mathbf{y} - \mu\mathbf{1}), \sigma^2 + \tau^2 - \gamma^T \Sigma^{-1} \gamma\right). \tag{20}$$

3.1 Definition of Prior Distributions

We use a *normal* prior for μ to allow this parameter to have positive or negative values and a *lognormal* prior for ϕ, σ^2 and τ^2 since these parameters can take only positive values. These distributions are simple to interpret and to tune to make them as non-informative as one may wish:

$$\mu \sim N(\mu_\mu, \sigma_\mu^2)$$
$$\phi \sim logN(\mu_\phi, \sigma_\phi^2)$$
$$\sigma^2 \sim logN(\mu_{\sigma^2}, \sigma_{\sigma^2}^2)$$
$$\tau^2 \sim logN(\mu_{\tau^2}, \sigma_{\tau^2}^2)$$

These priors are set quite non-informative to allow the posterior distributions to be solely influenced by the data. Hence, the variance parameter of the normal prior for μ, σ_μ^2, is set to 10^{40} and the variance parameters for *lognormal* priors, $\sigma_\phi^2, \sigma_{\sigma^2}^2$ and $\sigma_{\tau^2}^2$, are all set to 10^2 which are all very non-informative but still proper.

Based on the given prior distributions, the joint posterior distribution of the parameters is derived in Appendix B, where the full conditional distributions for ϕ, σ^2 and τ^2 are provided. Since these distributions are not known probability distributions, we need to use the Metropolis-Hastings algorithm to sample from the posterior distributions of these three parameters [31]. However, given the normal prior for μ, its full conditional can be written as a normal distribution (as shown in Appendix C); therefore, Gibbs sampling is used for this parameter [31].

3.2 The Bayesian EGO Algorithm

The algorithm can be summerized as follows:

- Design and run the initial experiments X to get $\mathbf{y}$ based on a design, e.g. Latin

Hypercube Sampling as in classic EGO [1]

- **While** stopping criteria (see Section 3.3) for Bayesian EGO algorithm is not met **do**
  1. run the MCMC and sample from the posterior distribution of $\Theta|\mathbf{y}$
  2. **while** stopping criteria for optimization procedure not met **do**
     i. at any new point $\mathbf{x}$, proposed by the optimization procedure, sample from the posterior predictive distribution of $y(\mathbf{x})|\mathbf{y}$
     ii. evaluate the Bayesian EI at $\mathbf{x}$ directly using equation (12)
  3. **end while**
  4. let the final solution of optimization procedure to be $\mathbf{x}^*$. Add $\mathbf{x}^*$ to the bottom of X
  5. run the function at $\mathbf{x}^*$ to get y^* and add it to the bottom of $\mathbf{y}$
- **End while**

3.3 Computational Details

In this section, we describe some implementation details of the MCMC sampling routine.

- As discussed in section 3.1, we use Gibbs sampling for μ and Metropolis-Hastings for the remaining three parameters of the covariance function, namely ϕ, σ^2 and τ^2. To determine the length of the Markov chains, a visual inspection of the trend plot, a plot of the variance and the mean of the variables and also a plot of the standard deviation of the mean were considered (see Appendix D for a sample of these monitoring plots). The chains should be long enough so that the trend plot shows stationarity and the other three plots are stabilized [32].

- In addition to inspection of MCMC chains, we calculate the Effective Sample Size (ESS) to determine the number of independent samples out of the generated samples. The ESS mainly depends on the autocorrelation structure within the generated samples: the more autocorrelated the samples are, the longer chains are required to achieve a given number of independent samples. The ESS was then used to perform "thinning" of the chains so that the remaining samples are no longer autocorrelated [32].

- To sample from the posterior distributions of the parameters ϕ, σ^2 and τ^2, we need to use the Metropolis-Hastings algorithm; therefore, a proposal distribution is required for each of them. We have used a log-normal distribution with $\mu_{proposal}$ equal to the logarithm of the parameter value at the previous MCMC iteration. Based on the the adaptive Metropolis-Hastings technique, proposed by Haario et al. [33], the $\sigma^2_{proposal}$ is adaptively changed along the MCMC chains. A good balance between the acceptance rate ($\approx 40\%$) and a rapid tail-off of the autocorrelation function [32] confirms the efficiency of the sampling.

- To calculate the MCMC variance of the parameters, batch means [32] were used given any possible autocorrelation exists in the Markov chains.

- The number of posterior samples was set equal to $\min(ESS, 1000)$ in all test functions. Note that the samples from the posterior distribution of the parameters are changing at each new point $\mathbf{x}$ proposed by the optimization routine.

- As noted later, the optimization routine is simply a search routine over a set of points defined by a fine grid or a space filling design [8] similar to Kleijnen et al. [2].

- The stopping criterion is either a predefined *maximum number of allowable iterations* or arriving at bayesian EI less than a predefined threshold value, $\exp(-20)$, the same as in [2].

4 Numerical Results

In this chapter we study the numerical behavior of the proposed bayesian EGO algorithm. First, the proposed model is justified in section 4.1. In section 4.2, the bayesian EGO algorithm is applied to optimize four well-known deterministic test functions. Section 4.3 provides some computational experience with the algorithm. Finally, in section 4.4, we apply our algorithm and the expected quantile improvement function to the optimization of a simulated stochastic inventory system.

4.1 Model Validation

The major objective of this section is to validate the proposed model in equation (17) for global optimization purposes. We suggested this model because of its mean and covariance structure simplicity which is valuable in the bayesian context since the computational complexity of the algorithm increases significantly with the number of parameters of the model.

Santer et al. [8] have done an extensive empirical study comparing different covariance functions and based on the results recommended to use the power exponential family ([8] p. 76). They also pointed out the potential of the Matern covariance family but mentioned that its implementation is computationally more expensive. For our bayesian EGO approach, we used an exponential covariance function which belongs to the power exponential family for isotropic processes with its power set to one. Fixing the power parameter in the power exponential family and not estimating

it is a common practice in the literature of global optimization, see [1], [2]. The power parameter determines the smoothness of the covariance function. We set the power p equal to one since we want the covariance function to be not too smooth. This helps the kriging predictor to suitably approximate even rapidly changing functions which can happen in practice.

Two remaining characteristics of the model that need to be addressed are the usage of isotropic covariance functions and fixed mean for which we used simulation analysis. We simulate Gaussian processes using the kriging prediction formula with certain properties (mean and covariance structure) as the objective function; then, we sample the simulated kriging function at some points (based on a predefined design). Finally, we use the bayesian EGO algorithm with the proposed model to find the global minimum of the underlying function.

We simulate two-dimensional ($\dim(\mathbf{x})=2$) non-convex functions using the kriging prediction formula (5) based on the model (2) similar to [34] and [35]. The general form of the mean structure is

$$\mu(\mathbf{x}) = b_0 + \mathbf{b}^T\mathbf{x} + \mathbf{x}^T B\mathbf{x} \tag{21}$$

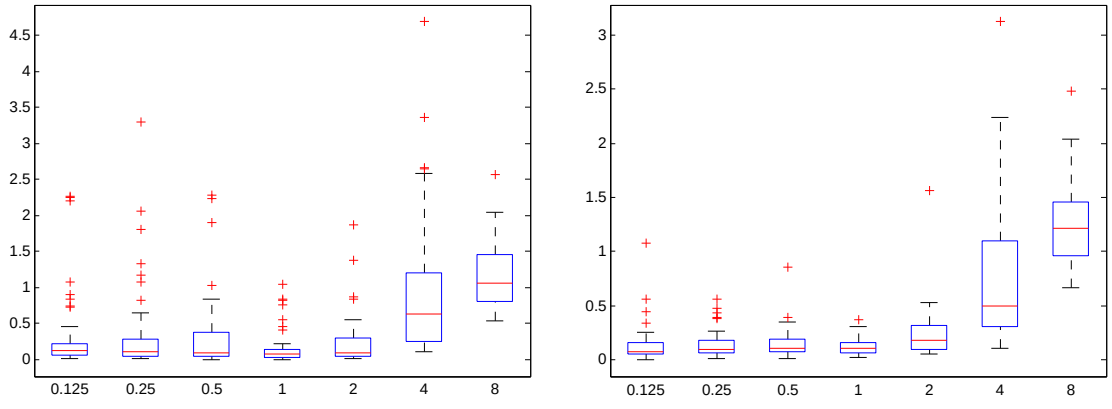
and the random process $z(\mathbf{x})$ is a zero mean stationary Gaussian process with the power exponential covariance function and power equal to one.

$$\mathrm{Cov}\{z(\mathbf{x}_i), z(\mathbf{x}_j)\} = \sigma^2 \exp\left(-\phi_1|\mathbf{x}_{i,1} - \mathbf{x}_{j,1}| - \phi_2|\mathbf{x}_{i,2} - \mathbf{x}_{j,2}|\right) \tag{22}$$

where ϕ_1 and ϕ_2 are the range parameters in the direction of x_1 and x_2 axes, respectively.

4.1.1 Sensitivity of the Bayesian EGO to Anisotropy

Our model assumes an isotropic covariance. To check for the sensitivity of our approach to anisotropy, we sampled from the kriging prediction formula with anisotropic correlation function (22) based on a Latin Hypercube Sampling design with 21 points on $[0, 10] \times [0, 10]$. In the covariance function (22), ϕ_1 is set to one and ϕ_2 is changed over range of seven different values $\{0.125, 0.25, 0.5, 1, 2, 4, 8\}$ (note that setting $\phi_2 = 1$ results into an isotropic process). Also, the variance parameter σ^2 is set to 1. Each of the seven combinations was replicated 50 times and the stopping criteria for the bayesian EGO algorithm in each replication is either to reach the expected improvement less than $\exp(-20)$ or to exceed the *maximum number of iterations* which is set to 40. The performance criteria considered are the Euclidean distance of the closest point found by the algorithm ($\mathbf{x}^*$) from the true global minimum ($\mathbf{x}^G$), $||\mathbf{x}^* - \mathbf{x}^G||$, and also the absolute difference of the minimum function value found by the algorithm (y_{min}) and the function value at the global minimum of the function (y^G), $|y_{min} - y^G|$. The results are shown in Figure 1.



(a) (b)

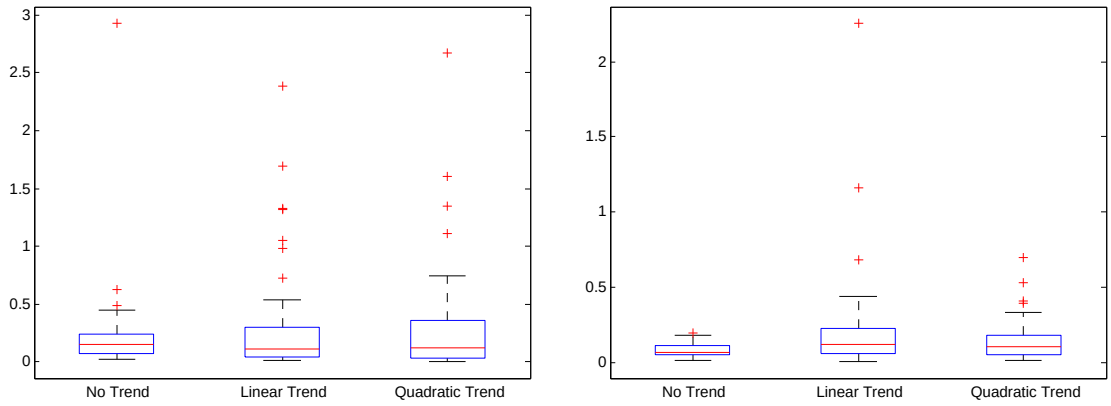
Figure 1: **(a)** Box plots of $||\mathbf{x}^* - \mathbf{x}^G||$ for seven different values of ϕ_2 **(b)** Box plots of $||y_{min} - y^G||$ for seven different values of ϕ_2

Figure 1 shows that only when the degree of anisotropy is very large (≥ 4), the closest distance to the global $||\mathbf{x}^* - \mathbf{x}^G||$ and the absolute difference of the minimum function value found and the function value at the global minimum $|y_{min} - y^G|$, significantly differ from zero. This shows that our bayesian EGO method is quite robust with respect to the isotropy assumption. As ϕ_2 gets larger and the process becomes uncorrelated in x_2 direction, the bayesian EGO's performance declines. In practice, such an extreme case are not common.

4.1.2 Performance of Bayesian EGO for the Optimization of Functions with a Non-Constant Mean (i.e., Trend)

We used a fixed mean μ in our model for the bayesian EGO as opposed to regression-type mean $\mathbf{x}^T\boldsymbol{\beta}$ in which the number of parameters is significantly higher (e.g. the number of parameters for a linear trend is the dimension of $\mathbf{x}$ plus one). We believe that the covariance structure can adapt the kriging predictor to even highly nonlinear functions with minimal number of parameters which is an important advantage in the bayesian optimization. To empirically test this issue, we sample from the kriging prediction formula based on LHS design with 21 points on $[0, 10] \times [0, 10]$ with three different mean structures as the three different scenarios. The first scenario which is the *no trend* scenario is implemented by setting b_0, $\mathbf{b}$ and B all equal to zero in (21). For the *linear trend* scenario $\mathbf{b} = [1, 1]^T$ while b_0 and B are set equal to zero. Finally, for the *quadratic trend*, B is the identity matrix of dimension two while b_0 and $\mathbf{b}$ are set to zero. The covariance function is similar to (22) with $\phi_1 = \phi_2 = 1$ (isotropic) and $\sigma^2 = 1$. For each scenario 50 replications are performed and the stopping criteria and the performance measures are as in Section 4.1.1. Figure 2 shows the results.

As we can see in Figure 2, the performance of the algorithm for the three different trend scenarios does not differ significantly. This means that the covariance structure can perfectly compensate the simplicity of the mean. A fixed mean structure is also



(a) (b)

Figure 2: **(a)** Box plots of $||\mathbf{x}^* - \mathbf{x}^G||$ for three different mean structures **(b)** Box plots of $||y_{min} - y^G||$ for three different mean structures

used in many dominant EGO papers e.g. [1] and [2].

4.2 Deterministic Test Functions

In this section, the performance of the bayesian EGO is studied using four test functions which are included in [2] and the results are compared with both the *classic EGO* and the *bootstrapped EGO*. The test functions include the one dimensional Forrester function, the two dimensional "six-hump camel-back" function, the three dimensional "Hartmann-3" function and the six dimensional "Hartmann-6" function.

The initial design X for each of the tests is a Latin Hypercube Sampling design in the function's domain [8]. This was the same design used for all the three algorithms (classic EGO, bootstrapped EGO and bayesian EGO). The stopping criterion is either reaching *maximum number of allowable iterations* (which is different for each of the four test functions and are set equal to those used in [2]) or attaining the expected improvement threshold which is set to $\exp(-20)$ (again, similar to [2]). Furthermore, to have a sense on variability, the algorithm is replicated 5 times for

each test function where each replication is starting from a different pseudorandom number seed. All of the calculations are done in MATLAB and the codes are available at *http://www2.ie.psu.edu/Castillo/research/EngineeringStatistics/software.htm*.

4.2.1 The Forrester Function

The first function that is evaluated is the one dimensional Forrester function:

$$y(x) = (6x - 2)^2 \sin(12x - 4) \quad 0 \leq x \leq 1$$

This function has one local minimum at $x^L = 0.01$ and one global minimum at $x^G = 0.7572$ where $y(x^G) = -6.0207$. The same initial design is used as [2] that is [0,0.5,1]. The optimization routine is a search over a grid with the step size equal to 0.01 in $(0, 1)$. Similar to [2], maximum number of allowable iterations is set to 8. Table 1 presents the results.

Table 1: Comparison of Classic EGO, Bootstrapped EGO and the proposed Bayesian EGO methods for the 1-D Forrester Function (Bootstrapped EGO results are included from [2])

	Rep.	x^*	$y(x^*)$	n^*	n_{total}	d
Class.EGO	1	0.76	-6.017	10	11	0.0028
Boots.EGO	1	0.76	-6.017	9	11	0.0028
	2	0.76	-6.017	10	11	0.0028
	3	0.76	-6.017	9	10	0.0028
	4	0.76	-6.017	10	10	0.0028
	5	0.76	-6.017	8	10	0.0028
Bayes.EGO	1	0.76	-6.017	8	11	0.0028
	2	0.76	-6.017	7	11	0.0028
	3	0.76	-6.017	9	11	0.0028
	4	0.78	-5.7282	10	11	0.0228
	5	0.76	-6.017	11	11	0.0028

Table 1 shows the coordinate of the optimal solution which is the closest point

to the global minimum (x^*) at each replication, the function value at that solution ($y(x^*)$), the iteration number which result in the optimal solution (n^*), the total number of iterations until stopping (n_{total})and finally the Euclidean distance between the optimal solution and the true global minimum (d). As we can see, the bayesian EGO method finds the true global minimum in four replications similar to the classic and the bootstrapped EGO. Furthermore, the bayesian EGO and the bootstrapped EGO seems to be almost the same in the speed of finding the optimum (based on n^*) while both of them are faster than the classic EGO. Notice that n^* and n_{total} include the initial design points, as well. Figure 3 illustrates the mean of the posterior predictive distribution, the variance of the posterior predictive distribution and also the bayesian EGO in $(0, 1)$ after 8 iterations of the algorithm. Note that the variance is higher at locations where the density of the observed data points are lower and vice versa. If there was a ninth iteration, it would be at the point which has the maximum bayesian EI in the plot.

4.2.2 The Six-Hump Camel-Back Function

The six-hump camel-back function is defined as

$$y(x_1, x_2) = 4x_1^2 - 2.1x_1^4 + x_1^6/3 + x_1x_2 - 4x_2^2 + 4x_2^4 \quad -2 \leq x_1 \leq 2, -1 \leq x_2 \leq 1$$

In the given domain the function has two global minima which are $\mathbf{x}_1^G = (0.089842, -0.712656)$ and $\mathbf{x}_2^G = (-0.089842, 0.712656)$ with the function value equal to -1.031628. Also, the function has two local minima.

Similar to [2], the initial design is a type of space filling design called maximin Latin Hypercube Sampling (LHS) with 21 points [8]. Furthermore, the optimization routine is a search over 200 candidate points generated from a maximin LHS design

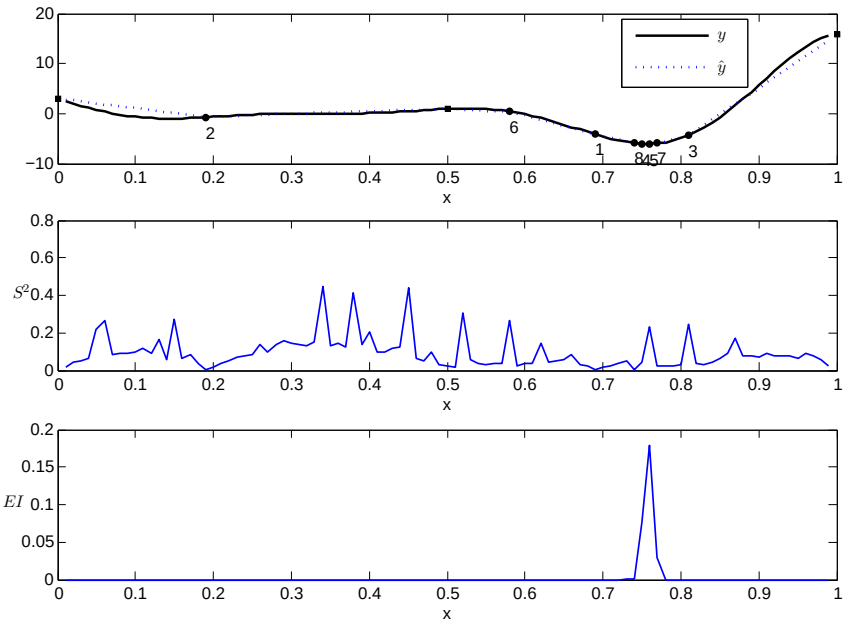


Figure 3: **Top:** The Forrester function (y) and the mean of the posterior predictive distribution ($\widehat{y}$). The squares are the initial design points and the round dots are the points suggested by bayesian EGO algorithm in one replication. **Middle:** The variance of the posterior predictive distribution. **Bottom:** The bayesian expected improvement

in the above domain. The maximum number of allowable iterations is set to 40 which is the same as [2]. Table 2 shows the results of the three methods.

The bayesian EGO method unanimously achieved better solutions compared to the classic EGO and the bootstrapped EGO considering that it uses all of its 40 allowable iterations. The distance of the optimal solutions from the global minima are lower for all of the replications compared to the other two approaches. However, both the classic and the bootstrapped EGO were faster than the bayesian EGO in finding their final solutions.

Figure 4 illustrates the contour plot of the six-hump camel-back function and its prediction by the mean of the posterior predictive distribution. The square black points are the 21 initial design points and the round red points are the suggested points by the proposed algorithm in one replication. These points are almost concentrated

Table 2: Comparison of Classic EGO, Bootstrapped EGO and the proposed Bayesian EGO methods for the 2-D Six-hump camel-back function(Bootstrapped EGO results are included from [2])

	Rep.	$\mathbf{x}^*$	$y(\mathbf{x}^*)$	n^*	n_{total}	d
Class.EGO	1	(-0.0302,0.7688)	-0.9863	31	41	0.0819
Boots.EGO	1	(0.0302,-0.7688)	-0.9863	29	43	0.0819
	2	(-0.0302,0.7688)	-0.9863	29	41	0.0819
	3	(-0.0302,0.7688)	-0.9863	29	42	0.0819
	4	(0.0302,-0.7688)	-0.9863	29	42	0.0819
	5	(0.0302,-0.7688)	-0.9863	29	43	0.0819
Bayes.EGO	1	(-0.0971,0.7333)	-1.0280	51	61	0.0219
	2	(0.0864,-0.7256)	-1.0301	61	61	0.0134
	3	(-0.0926,0.7065)	-1.0313	46	61	0.0067
	4	(0.0980,-0.7051)	-1.0308	32	61	0.0111
	5	(0.1086,-0.7188)	-1.0301	38	61	0.0198

around the two global minima which confirms the capability of the algorithm to locate both of the global minima.

Figure 5 shows the contours of the bayesian expected improvement for the six-hump camel-back function. Notice that the expected improvement function is maximized around the two global minima.

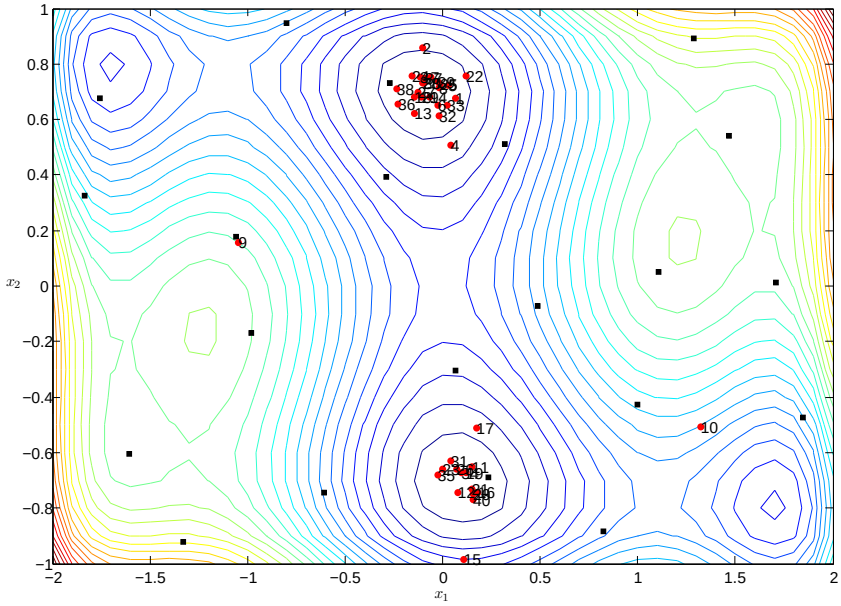
4.2.3 The Hartmann-3 Function

The Hartmann-3 is a three dimensional function defined as

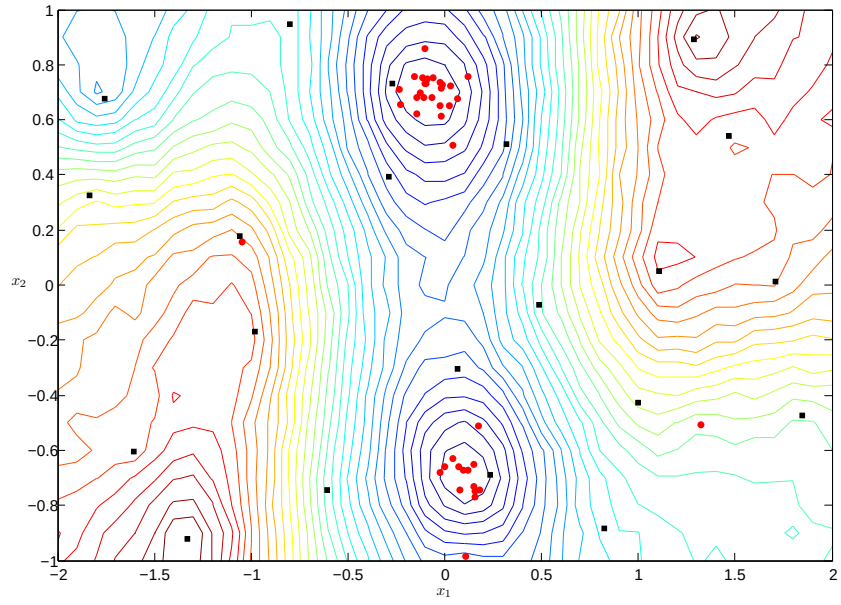
$$y(x_1, x_2, x_3) = -\sum_{i=1}^{4} \alpha_i \exp\left[-\sum_{j=1}^{3} A_{ij}(x_j - P_{ij})^2\right] \quad 0 \le x_i \le 1, i = 1, 2, 3$$

where $\alpha = (1.0, 1.2, 3.0, 3.2)$ and A_{ij} and P_{ij} are given in Table 3. The function has one global minimum at $\mathbf{x}^G = (0.114614, 0.555649, 0.852547)$ with function value equal to -3.86278 and also three local minima.

The initial design is a maximin LHS design with 30 points. To perform the



(a)



(b)

Figure 4: **(a)** Contour plot of the six-hump camel-back function. The square (black) points show the initial LHS design and the round (red) points are the points suggested by the bayesian EGO algorithm in one replication. The number beside each round (red) point is the iteration number of that solution **(b)** Contour plot of the predicted six-hump camel-back function through the mean of the posterior predictive distribution.

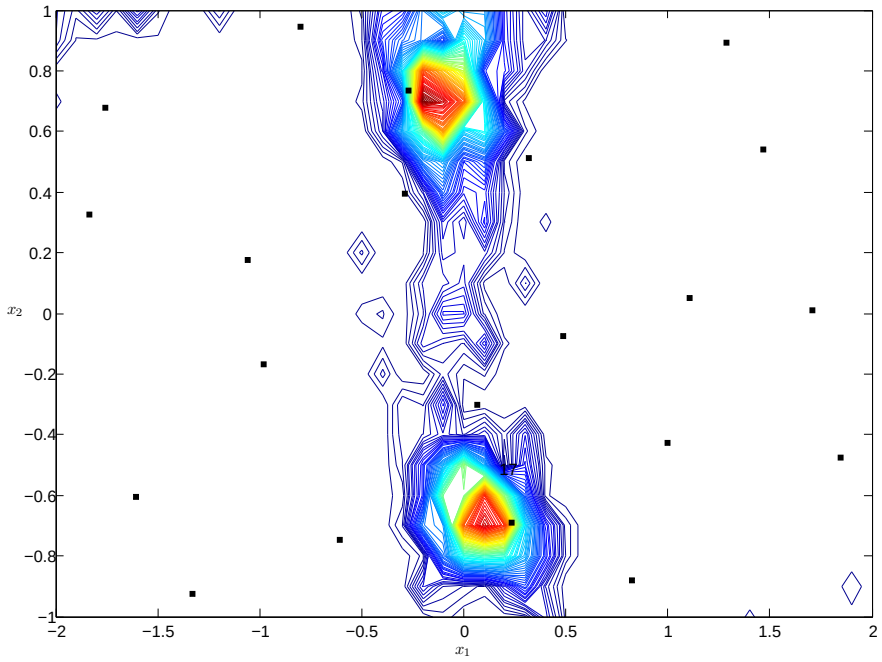


Figure 5: Contour plot of the bayesian expected improvement for the six-hump camel-back function. The square (black) points are the initial LHS design.

Table 3: parameters A_{ij} and P_{ij} for the Hartmann-3 Function

A_{ij}			P_{ij}		
3.0	10	30	0.36890	0.11700	0.26730
0.1	10	35	0.46990	0.43870	0.74700
3.0	10	30	0.10910	0.87320	0.55470
0.1	10	35	0.03815	0.57430	0.88280

optimization, the bayesian EGO is evaluated over a maximin LHS design with 300 points and the maximum is the final solution. Furthermore, the maximum number of iterations is set to 35 (all of the settings are similar to [2]). Table 4 shows the results for the Hartmann-3 function.

The function values at the optimal solutions of all of the replications of the bayesian EGO method are lower than those of the classic and the bootstrapped EGO. However, both the classic and the bootstrapped EGO found their optimal solutions faster than the bayesian EGO method except for replication 3.

Table 4: Comparison of the Classic EGO, the Bootstrapped EGO and the Bayesian EGO methods for the 3-D Hartmann-3 function(The Bootstrapped EGO results are included from [2])

	Rep.	$\mathbf{x}^*$	$y_{\mathbf{x}^*}$	n^*	n_{total}	d
Class.EGO	1	(0.2088,0.5465,0.8767)	-3.7956	44	65	0.0977
Boots.EGO	1	(0.2088,0.5465,0.8767)	-3.7956	34	65	0.0977
	2	(0.2088,0.5465,0.8767)	-3.7956	34	65	0.0977
	3	(0.2088,0.5465,0.8767)	-3.7956	41	65	0.0977
	4	(0.2088,0.5465,0.8767)	-3.7956	34	65	0.0977
	5	(0.2088,0.5465,0.8767)	-3.7956	44	65	0.0977
Bayes.EGO	1	(0.0780,0.5615,0.8628)	-3.8510	35	65	0.0385
	2	(0.2752,0.5618,0.8643)	-3.8330	60	65	0.1611
	3	(0.1639,0.5637,0.8431)	-3.8501	34	65	0.0509
	4	(0.1388,0.5818,0.8553)	-3.8381	53	65	0.0357
	5	(0.0083,0.5598,0.8506)	-3.8548	61	65	0.1064

Figure 6 shows the points proposed by the bayesian EGO algorithm in one replication (round red) and the global minimum (star shaped green).

4.2.4 The Hartmann-6 Function

The last test function which was evaluated is the Hartmann-6 function with six variables. It is defined as

$$y(x_1, ..., x_6) = -\sum_{i=1}^{4} c_i \exp\left[-\sum_{j=1}^{6} \alpha_{ij}(x_j - p_{ij})^2\right] \quad 0 \leq x_i \leq 1, i = 1, ..., 6$$

where $c = (1.0, 1.2, 3.0, 3.2)$ and α_{ij} and p_{ij} are given in Table 5. This function has a global minimum at $\mathbf{x}^G = (0.2017, 0.1500, 0.4768, 0.2753, 0.3116, 0.6573)$ with function value equal to -3.32237 and five local minima.

The initial design is a maximin LHS with 50 points. Furthermore, the set of candidate points in the search optimization routine is from a maximin LHS design with 500 points in the function space. Finally, the maximum number of allowable

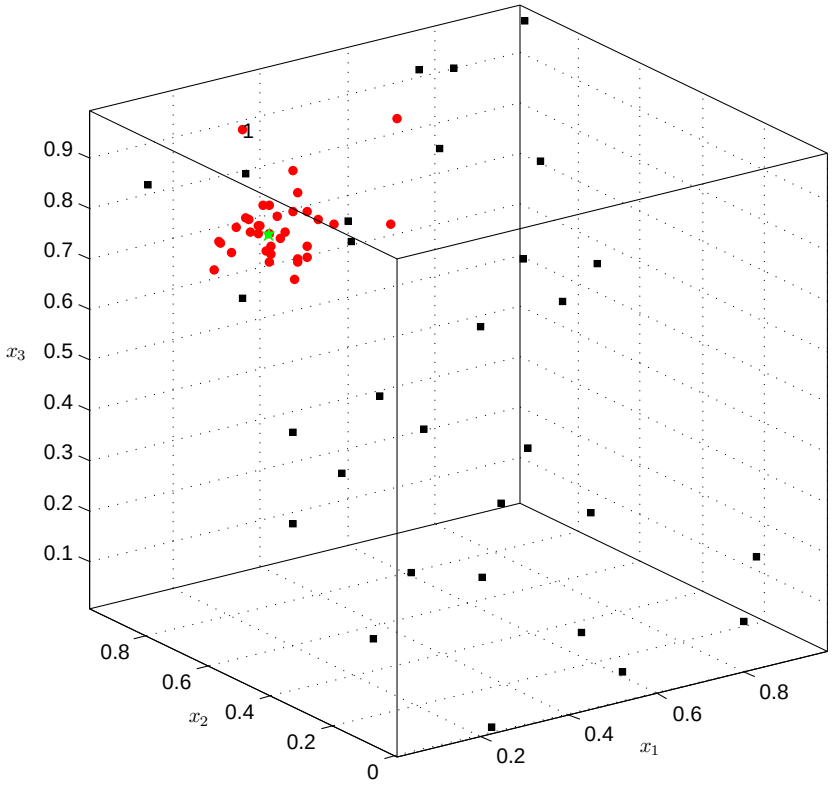


Figure 6: Square (black) points are the initial LHS design, round (red) points are points proposed by the bayesian EGO algorithm in one replication and the star (green) shows the global minimum of the Hartmann-3 function.

Table 5: Parameters α_{ij} and p_{ij} of Hartmann-6 Function

α_{ij}	10.0	3.0	17.0	3.5	1.7	8.0
	0.05	10.0	17.0	0.1	8.0	14.0
	3.0	3.5	1.7	10.0	17.0	8.0
	17.0	8.0	0.05	10.0	0.1	14.0
p_{ij}	0.1312	0.1696	0.5569	0.0124	0.8283	0.5886
	0.2329	0.4135	0.8307	0.3736	0.1004	0.9991
	0.2348	0.1451	0.3522	0.2883	0.3047	0.6650
	0.4047	0.8828	0.8732	0.5743	0.1091	0.0381

iterations is set to 50 (all of these settings are similar to [2]). Table 6 compares the results of the bayesian EGO for the Hartmann-6 function versus the classic and the bootstrapped EI methods.

Table 6: Comparison of the Classic EGO, the Bootstrapped EGO and the Bayesian EGO methods for the 6-D Hartmann-6 function(the Bootstrapped EGO results are included from [2])

	Rep.	$\mathbf{x}^*$	$y(\mathbf{x}^*)$	n^*	n_{total}	d
Class.EGO	1	(0.3535,0.8232,0.8324,0.4282,0.1270,0.0013)	-2.3643	79	100	1.0442
Boots.EGO	1	(0.3535,0.8232,0.8324,0.4282,0.1270,0.0013)	-2.3643	92	100	1.0442
	2	(0.3535,0.8232,0.8324,0.4282,0.1270,0.0013)	-2.3643	89	100	1.0442
	3	(0.3535,0.8232,0.8324,0.4282,0.1270,0.0013)	-2.3643	78	100	1.0442
	4	(0.3535,0.8232,0.8324,0.4282,0.1270,0.0013)	-2.3643	86	100	1.0442
	5	(0.3535,0.8232,0.8324,0.4282,0.1270,0.0013)	-2.3643	92	100	1.0442
Bayes.EGO	1	(0.1570,0.1386,0.4755,0.3421,0.2334,0.6297)	-2.8681	94	100	0.1160
	2	(0.2113,0.0702,0.4313,0.3100,0.2410,0.6060)	-2.9112	83	100	0.1317
	3	(0.0933,0.2175,0.3270,0.2391,0.3446,0.5733)	-2.6851	95	100	0.2196
	4	(0.1321,0.1530,0.5626,0.3280,0.2955,0.6399)	-3.0836	70	100	0.1246
	5	(0.2290,0.0795,0.2978,0.2877,0.2405,0.6839)	-2.8464	70	100	0.2091

Note that the $y(\mathbf{x}^*)$ for the bayesian EGO algorithm is lower in all of the replications compared to the bootstrapped and the classic EGO methods. Furthermore, the bayesian EGO is faster in two replications (replications 4 and 5) in finding its final solution compared to the bootstrapped and the classic EGO methods.

4.3 Computation Time

In this section, we give insight on the computational time of the bayesian EGO algorithm. Duration of each run of the bayesian EGO algorithm depends mainly on the following factors: 1. Number of input variables - dimension (Dim) 2. Initial design size (IDS) 3. Maximum number of allowable iterations (MNAI) 4. MCMC Chains length (CL) 5. Number of the posterior samples used to evaluate the posterior predictive distribution at a given location (NPS). Table 7 shows some statistics on

computational time of a single iteration for the four test functions on a 3.60 GHz Intel pentium processor with 4.00 GB of RAM.

Table 7: Computational time of a single iteration for the four test functions on a 3.60 GHz Intel pentium processor with 4.00 GB of RAM

						Iteration Time (sec)			
Function	Dim	IDS	MNAI	CL	NPS	Min	Mean	Max	Std.
Forrester	1	3	8	1e4	200	57.56	58.85	61.04	1.48
Six-Hump	2	21	40	1e4	200	100.57	107.15	113.62	5.80
Hartmann-3	3	30	35	1e4	200	114.10	122.68	130.51	5.97
Hartmann-6	6	51	50	3e4	1e3	600.62	659.49	732.24	55.42

The iteration times increase as the size of the matrix X increases either due to the number of observations or the dimension of $\mathbf{x}$. The iteration times are the mean iteration time within one replication of the bayesian EGO algorithm and the provided statistics are over five different replications. Notice that the MCMC chains length was increased for the Hartmann-6 test function due to the existence of higher autocorrelation.

4.4 Bayesian Quantile EGO to a Stochastic Inventory Simulation

We now presents our bayesian EGO algorithm for the optimization of a simulated stochastic inventory system. The inventory model is taken from Law and Kelton [36] and the problem consists of finding the optimal reorder point (s) and the maximal holding quantity (S) in an (s, S) inventory policy. The objective function is the expected total cost per month which is the sum of the ordering cost, the holding cost and the shortage cost. Under an (s, S) policy, a company reviews its inventory at the beginning of each month and decides how much to order. Based on the (s, S) policy the order quantity is

$$Z = \begin{cases} S - I & \text{if } I < s \\ 0 & \text{if } I \geq s \end{cases}$$

where I is the inventory level and Z is the order quantity at the beginning of each month. The time between demands are i.i.d. exponential random variables with mean of 0.1 month. Furthermore, the size of the demands, D, are i.i.d. random variable with the following probability mass function:

$$D = \begin{cases} 1 & \text{w.p. } \frac{1}{6} \\ 2 & \text{w.p. } \frac{1}{3} \\ 3 & \text{w.p. } \frac{1}{3} \\ 4 & \text{w.p. } \frac{1}{6} \end{cases}$$

Finally, the supplier's *lead time* is a uniform random variable between 0.5 and 1 month. Each order has a fixed setup cost of $K = \$32$ and a linear incremental cost of $i = \$3$ per item (if the order quantity is zero then the setup cost is also zero). The holding cost is $h = \$1$ per item per month and the shortage cost is $p = \$5$ per item per month.

The literature on finding optimal (s, S) inventory policies is extensive, and mostly solve the problem using the dynamic programming techniques both in finite and infinite time horizon. Here, we want to find the optimal policy in infinite time horizon such that the optimal reorder point (s) and the maximal holding quantity (S) do not change over time. Given that the lead time is stochastic, there does not exist any theoretical result for an optimal (s, S) inventory policy. Actually, this is the reason why the simulation optimization procedures are so much popular for the inventory models with stochastic lead time. The closest theoretical work that was found for this inventory problem is by Ehrhardt which is based on power approximation, [37] and [38]. We compare our results with this method.

The initial design that we used is a maximin LHS with 21 points and the maximum

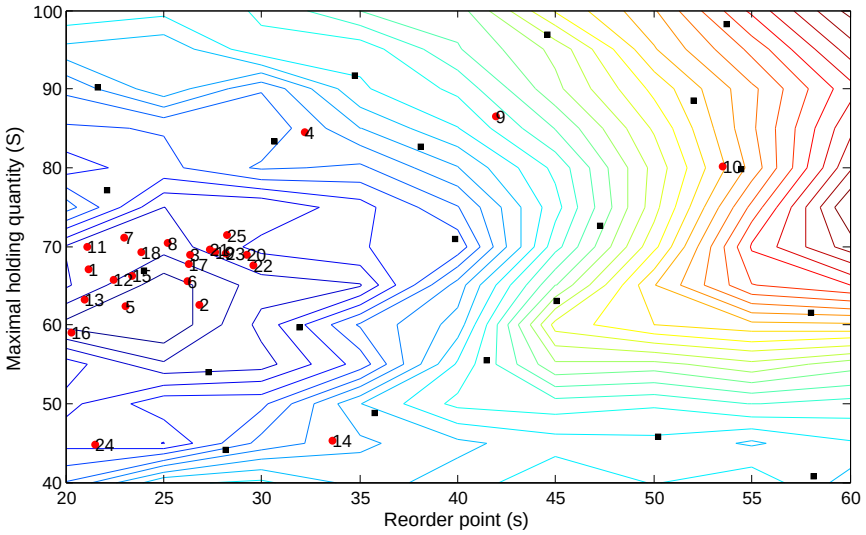
number of iterations is set to 25. A realization of the stochastic process along with the bayesian prediction through the mean of the posterior predictive distribution is illustrated in Figure 7. The posterior predictive mean seems to be able to predict the stochastic function to a reasonable extent.

The inventory simulation is stochastic; hence, we decided to implement the Expected Quantile Improvement (EQI) of [25] at each iteration of EGO algorithm with the quantile (β) set equal to 0.75. Having the posterior predictive distribution at any point $\mathbf{x}$, we can easily calculate the conditional expectation of any function of the random variable with respect to the distribution of $y(\mathbf{x})|\mathbf{y}$ which is an important advantage of the bayesian approach. Herein, the function of the random variable is the EQI as shown in (15) which we calculate at each iteration of the bayesian EGO algorithm to find the next point to evaluate the simulation.

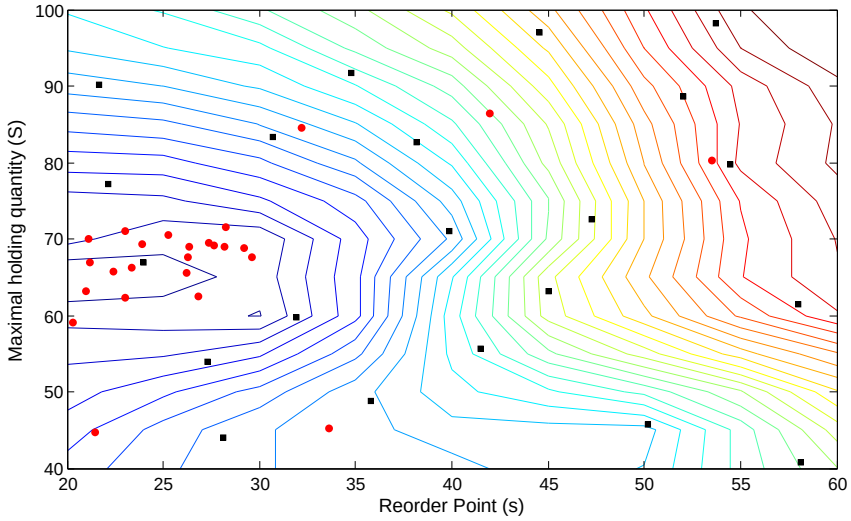
The optimal solutions based on the theoretical approximation method of Ehrhardt and our proposed bayesian EGO method are provided in Table 8. Furthermore, the inventory simulation model was ran 1000 times for each solution and the corresponding mean, standard deviation and 95% confidence intervals are reported. The results propose that the bayesian EGO method came up with a solution with lower inventory costs compare to the Ehrhardt method.

Table 8: The mean, the standard deviation and the 95% CI for total inventory costs based on 1000 replications of the inventory simulation model at the optimal solutions

Method	(s^*, S^*)	Mean	Std.	95% CI
Ehrhardt	(39.72,79.03)	125.99	2.35	(125.85,126.14)
Bayesian EGO	(25.06,59.12)	119.23	3.24	(119.03,119.43)



(a)



(b)

Figure 7: **(a)** Contour plot of a realization of the stochastic cost function for the inventory model example. The square (black) points are the initial LHS design and the round (red) points are the points suggested by the bayesian EGO algorithm. The number beside each red dot is the iteration number of that solution. **(b)** Contour plot of the predicted simulation model through the mean of the posterior predictive distribution

5 Conclusions and Further Work

In this dissertation a fully bayesian implementation of the EGO method was presented which considers the uncertainty in the parameters. Instead of using a plug-in estimator for the kriging variance which underestimates the true variance, or using a bootstrapped estimate of this variance which entails repeated difficult optimizations, a bayesian expected improvement was proposed and embedded within the EGO algorithm for optimization of unknown and non-convex functions.

The simple model used in this approach was first validated through simulation. The proposed model assumes the underlying process to be isotropic and to have a constant mean. The sensitivity of the model was first analyzed for the cases where the underlying process in not isotropic with different level of anisotropy. The bayesian EGO algorithm performed quite robust to the isotropy assumption. Furthermore, performance of the algorithm was evaluated for a process without a constant mean. We assumed a linear and a quadratic mean structure for the simulating function and then used our method to optimize it. The bayesian EGO showed that the covariance structure can perfectly compensate for the constant mean and predict the underlying function.

Next, the performance of the proposed approach was compared with the classic and the bootstrapped EGO methods for four different deterministic test functions from the literature. The function values at the optimal solutions, the distance of the optimal solutions from the true global optimum(s) and the speed of achieving the optimal solutions were then compared across the three different methods. In general,

the bayesian EGO method found solutions with better function values and locations closer to the global optimum(s) especially for higher dimensional functions.

However, on average, the bayesian EGO was slower in finding the optimal solution compared to the classic and the bootstrapped EGO methods. Speeding up the bayesian EGO is therefore a matter of further research. We suggest using Sequential Monte Carlo (SMC) methods [39] to faster sample from the posterior distributions in this iterative procedure. Currently, each time that a new solution and its corresponding function value are added to the matrix X and the vector $\mathbf{y}$, the Monte Carlo Markov Chains are rerun to sample from the posterior distributions in the next iteration. This is a rather lengthy computation and can be significantly shortened by SMC techniques.

Furthermore, the bayesian EGO approach was implemented for a stochastic inventory cost function by calculating the Expected Quantile Improvement (EQI) of [25] to find the optimal reorder point (s) and the maximal holding quantity (S) in an (s, S) inventory policy. This is a major advantage of the bayesian methods which provide the posterior predictive distributions. The expectation of any function of the random variable with respect to the conditional distribution of $y(\mathbf{x})|\mathbf{y}$ can easily be calculated; hence, with minimal effort we can calculated EQI instead of EI for optimization of a stochastic simulation code. The performance of the proposed approach was compared with the theoretical power approximation method of Ehrhardt [37]. The results showed the advantage of the bayesian EGO method in optimizing the stochastic simulation model as compared to the theoretical power approximation method.

Bibliography

[1] Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functios," *Journal of Global Optimization*, volume 13, no. 4, pp. 455–492, 1998

[2] Kleijnen, Jack P. C., van Beers, Wim, and van Nieuwenhuyse, Inneke, "Expected improvement in efficient global optimization through bootstrapped kriging," *Journal of Global Optimization*, volume 54, pp. 59–73, 2012

[3] Zhigljavsky, Anatoly and Zilinskas, Antanas, *Stochastic Global Optimization*, Springer, ISBN 9780387747408, 2008

[4] Simpson, T. J., Koch, P., and Allen, J., "Metamodels for computer-based engineering design: Survey and recommendations," *Engineering with Computers*, volume 17, pp. 129–150, 2001

[5] Krige, Danie G., "A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand," Technical report

[6] Matheron, Georges, "Principles of Geostatistics," *Economic Geology*, volume 58, pp. 1246–1266, 1963

[7] Cressie, Noel A. C., *Statistics for spatial data*, John Wiley and Sons, ISBN 0471002550, 1993

[8] Santner, Thomas J., Williams, Brian J., and Notz, William, *The Design and Analysis of Computer Experiments*, Springer, ISBN 0387954201, 2003

[9] Sacks, Jerome, Welch, William J., Mitchell, Toby J., and Wynn, Henry P., "Design and Analysis of Computer Experiments," *Statistical Science*, volume 4, no. 4, pp. 409–435, 1989

[10] Ginsbourger, David and Riche, Rodolphe Le, "Towards GP-based optimization with finite time horizon," *HAL EMSE*, volume 00424309, 2009

[11] Kushner, Harold J., "A versatile Stochastic Model of a Function of unknown and Time Varying Form," *Journal of Mathematical Analysis and Applications*, volume 5, pp. 15–167, 1962

[12] Mockus, J., Tiesis, V., and Zilinskas, A., *Towards Global Optimisation*, chapter The application of bayesian methods for seeking the extremum, North Holland, Amsterdam, pp. 117–129, 1978

[13] Zilinskas, Antanas, "Optimization of One-Dimensional Multimodal Functions," *Journal of the Royal Statistical Society Applied Statistics*, volume 27, no. 3, pp. 367–375, 1978

[14] Zilinskas, Antanas, "On Statistical Models for Multimodal Optimization," *Series Statistics*, volume 9, no. 2, pp. 255–266, 1978

[15] Zilinskas, Antanas, "One-Step Bayesian Method for the Search of the Optimum of One-Variable Functions," *Cybernetics*, , no. 1, pp. 139–144, 1975

[16] Zilinskas, Antanas, "Axiomatic Characterization of a Global Optimization Algorithm and Investigation of its Search Strategies," *Operations Research*, volume 4, pp. 35–39, 1985

[17] Torn, Aimo and Zilinskas, Antanas, *Lecture Notes in Computer Science*, Springer, ISBN 3540508716, 1987

[18] Williams, B. J., Santner, T. J., and Notz, W. I., "Sequential design of computer experiments to minimize integrated response functions," *Statistica Sinica*, volume 10, pp. 1133–1152, 2000

[19] Williams, Brian J., Santner, Thomas J., Notz, William I., and Lehman, Jeffrey S., "Sequential design of computer experiments for constrained optimization," *Statistical Modeling and Regression Structures*, pp. 449–472, 2010

[20] del Castillo, Enrique and Santiago, Eduardo, "A matrix-T approach to the sequntial design of optimization experiments," *IIE Transactions*, volume 43, pp. 54–68, 2011

[21] Sasena, M. J., Papalambros, P., and Goovaerts, P., "Exploration of metamodeling sampling criteria for constrained global optimization," *Engineering optimization*, volume 34, pp. 263–278, 2002

[22] Jones, Donald R., "A Taxonomy of Global Optimization Methods Based on Response Surfaces," *Journal of Global Optimization*, volume 21, pp. 345–383, 2001

[23] Huang, D., Allen, T. T., Notz, W. I., and Zeng, N., "Global optimization of stochastic black-box systems via sequential kriging meta-models," *Journal of Global Optimization*, volume 34, pp. 441–466, 2006

[24] Roustant, Olivier, Ginsbourger, David, and Deville, Yves, "DiceKriging, DiceOptim: Two R Pachages for Analysis of Computer Experiments by Kriging-Based Metamodeling and Optimization," *Journal of Statistical Software*, volume 51, no. 1, 2012

[25] Pichney, Victor, Ginsbourger, David, Richet, Yann, and Caplin, Gregory, "Quantile-Based Optimization of Noisy Computer Experiments with Tunable Precision," *Technometrics*, volume 55, no. 1, pp. 2–13, 2013

[26] den Hertog, D., Kliejnen, Jack P. C., and Siem, A. Y. D., "The correct Kriging variance estimated by bootstrapping," *Journal of Operational Research Society*, volume 57, no. 4, pp. 400–409, 2006

[27] Sjostedt De Luna, Sara and Young, A., "The bootstrap and kriging prediction intervals," *Scandinavian Journal of Statistics*, volume 30, pp. 175–192, 2003

[28] Benassi, Romain, Bect, Julien, and Vazquez, Emmanuel, "Robust Gaussian Process-Based Global Optimization Using a Fully Bayesian Expected Improvement Criterion," *Learning and Intelligent Optimization*, volume 6683 of *Lecture Notes in Computer Science*, Springer, pp. 176–190, 2011

[29] Handcock, Mark S. and Stein, Michael L., "A Bayesian Analysis of Kriging," *Technometrics*, volume 35, no. 4, pp. 403–410, 1993

[30] Anderson, Theodore W., *An Introduction to Multivariate Statistical Analysis*, Wiley Series in Probability and Statistics, Wiley, ISBN 978-0-471-36091-9, 2003

[31] Carlin, Bradley P. and Louis, Thomas A., *Bayesian methods for data analysis*, CRC Press, ISBN 9781584886976, 2009

[32] Robert, Christian and Casella, George, *Introducing Monte Carlo Methods with R*, Springer, ISBN 9781441915757, 2010

[33] Haario, Heikki, Saksman, Eero, and Tamminen, Johanna, "An adaptive Metropolis algorithm," *Bernouli*, volume 7, no. 2, pp. 223–242, 2001

[34] Trosset, Michael W., "The Krigifier: A Procedure for Generating Pseudorandom Nonlinear Objective Functions for Computational Experiments," ICASE Interim

Report 35, Institude for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, Virginia, February 1999

[35] Trosset, Michael W. and Padula, Anthony D., "Designing and Analyzing Computational Experiments for Global Optimization," Technical report, Department of Mathematics, College of William and Mary, 2000

[36] Law, Averill M. and Kelton, David, *Simulation modeling and analysis*, McGraw-Hill, ISBN 0070592926, 2000

[37] Ehrhardt, Richard, "(s,S) policies for dynamic inventory model with stochastic lead times," *Operations Research*, volume 32, no. 1, pp. 121–132, 1984

[38] Ehrhardt, Richard, "The power approximation for computing (s,S) inventory policies," *Management Science*, volume 25, no. 8, 1979

[39] Doucet, Arnaud, de Freitas, Nando, and Gordon, Neil, *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science, Springer, ISBN 0-387-95146-6, 2001

[40] Banerjee, Sudipto, Carlin, Bradley P., and Gelfand, Alan E., *Hierarchical Modeling and Analysis for Spatial Data*, CRC Press, ISBN 1-58488-410-X, 2004

A Derivation of the Expected Improvement Formula

Below you can find the derivation for the Expected Improvement closed form formula (13). For simplicity, we have replaced $y(\mathbf{x})$, $\widehat{y}(\mathbf{x})$ and $s(\mathbf{x})$ with y, $\widehat{y}$ and s, respectively.

$$
\begin{aligned}
E[I(\mathbf{x})] &= E[\max(y_{min} - y(\mathbf{x}), 0)] \\
&= \int \max(y_{min} - y) f(y) dy \\
&= \int_{-\infty}^{y_{min}} (y_{min} - y) f(y) dy \\
&= y_{min} \int_{-\infty}^{y_{min}} f(y) d(y) - \int_{-\infty}^{y_{min}} y f(y) dy \\
&= y_{min} \Phi\left(\frac{y_{min} - \widehat{y}}{s}\right) - \frac{1}{\sqrt{2\pi} s} \underbrace{\int_{-\infty}^{y_{min}} y \exp\left(\frac{(y - \widehat{y})^2}{-2s^2}\right) dy}_{A}
\end{aligned}
$$

where,

$$
\begin{aligned}
A &= \int_{-\infty}^{y_{min}} (y-\widehat{y}) \exp\left(\frac{(y-\widehat{y})^2}{-2s^2}\right) dy + \widehat{y}\int_{-\infty}^{y_{min}} \exp\left(\frac{(y-\widehat{y})^2}{-2s^2}\right) dy \\
&= -s^2 \int_{-\infty}^{y_{min}} \frac{(y-\widehat{y})}{-s^2} \exp\left(\frac{(y-\widehat{y})^2}{-2s^2}\right) dy + \widehat{y}\int_{-\infty}^{y_{min}} \exp\left(\frac{(y-\widehat{y})^2}{-2s^2}\right) dy \\
&= -s^2 \left[\exp\left(\frac{(y-\widehat{y})^2}{-2s^2}\right)\right]_{-\infty}^{y_{min}} + \widehat{y}\left(\sqrt{2\pi}s\Pr(y \le y_{min})\right) \\
&= -s^2 \exp\left(\frac{(y_{min}-\widehat{y})^2}{-2s^2}\right) + \widehat{y}\left(\sqrt{2\pi}s\Phi\left(\frac{y_{min}-\widehat{y}}{s}\right)\right) \\
&= -s^2\sqrt{2\pi}\underbrace{\frac{1}{\sqrt{2\pi}} \exp\left(\frac{\left(\frac{y_{min}-\widehat{y}}{s}\right)^2}{-2}\right)}_{\phi\left(\frac{y_{min}-\widehat{y}}{s}\right)} + \widehat{y}\sqrt{2\pi}s\Phi\left(\frac{y_{min}-\widehat{y}}{s}\right) \\
&= -\sqrt{2\pi}s^2\phi\left(\frac{y_{min}-\widehat{y}}{s}\right) + \sqrt{2\pi}\widehat{y}s\Phi\left(\frac{y_{min}-\widehat{y}}{s}\right)
\end{aligned}
$$

Hence,

$$
\begin{aligned}
E[I(\mathbf{x})] &= y_{min}\Phi\left(\frac{y_{min}-\widehat{y}}{s}\right) - \frac{1}{\sqrt{2\pi}s}\left(-\sqrt{2\pi}s^2\phi\left(\frac{y_{min}-\widehat{y}}{s}\right) + \sqrt{2\pi}\widehat{y}s\Phi\left(\frac{y_{min}-\widehat{y}}{s}\right)\right) \\
&= y_{min}\Phi\left(\frac{y_{min}-\widehat{y}}{s}\right) + s\phi\left(\frac{y_{min}-\widehat{y}}{s}\right) - \widehat{y}\Phi\left(\frac{y_{min}-\widehat{y}}{s}\right) \\
&= (y_{min}-\widehat{y})\Phi\left(\frac{y_{min}-\widehat{y}}{s}\right) + s\phi\left(\frac{y_{min}-\widehat{y}}{s}\right)
\end{aligned}
$$

B Posterior and Full Conditional Distributions

The posterior distribution is

$$
\begin{aligned}
p(\mu, \phi, \sigma^2, \tau^2|\mathbf{y}) \propto (2\pi)^{-n/2}|\Sigma|^{-1/2} \exp\left(\frac{-1}{2}(\mathbf{y}-\mu\mathbf{1})^T\Sigma^{-1}(\mathbf{y}-\mu\mathbf{1})\right) \\
\times (2\pi\sigma_\mu^2)^{-1/2} \exp\left(\frac{(\mu-\mu_\mu)^2}{-2\sigma_\mu^2}\right) \\
\times \phi^{-1}(2\pi\sigma_\phi^2)^{-1/2} \exp\left(\frac{(\ln\phi-\mu_\phi)^2}{-2\sigma_\phi^2}\right) \\
\times (\sigma^2)^{-1}(2\pi\sigma_{\sigma^2}^2)^{-1/2} \exp\left(\frac{(\ln\sigma^2-\mu_{\sigma^2})^2}{-2\sigma_{\sigma^2}^2}\right) \\
\times (\tau^2)^{-1}(2\pi\sigma_{\tau^2}^2)^{-1/2} \exp\left(\frac{(\ln\tau^2-\mu_{\tau^2})^2}{-2\sigma_{\tau^2}^2}\right)
\end{aligned}
$$

which is basically the likelihood multiplied by the priors for μ, ϕ, σ^2 and τ^2. Given the joint posterior distribution of the parameters, the full conditional distribution of ϕ, σ^2 and τ^2 can be written as follows:

$$
p(\phi|\mu, \sigma^2, \tau^2, \mathbf{y}) = |\Sigma|^{-1/2} \exp\left(\frac{-1}{2}(\mathbf{y}-\mu\mathbf{1})^T\Sigma^{-1}(\mathbf{y}-\mu\mathbf{1})\right)(\phi)^{-1} \exp\left(\frac{(\ln\phi-\mu_\phi)^2}{-2\sigma_\phi^2}\right)
$$

$$p(\sigma^2|\mu,\phi,\tau^2,\mathbf{y}) = |\Sigma|^{-1/2}\exp\left(\frac{-1}{2}(\mathbf{y}-\mu\mathbf{1})^T\Sigma^{-1}(\mathbf{y}-\mu\mathbf{1})\right)(\sigma^2)^{-1}\exp\left(\frac{(\ln\sigma^2-\mu_{\sigma^2})^2}{-2\sigma^2_{\sigma^2}}\right)$$

$$p(\tau^2|\mu,\phi,\sigma^2,\mathbf{y}) = |\Sigma|^{-1/2}\exp\left(\frac{-1}{2}(\mathbf{y}-\mu\mathbf{1})^T\Sigma^{-1}(\mathbf{y}-\mu\mathbf{1})\right)(\tau^2)^{-1}\exp\left(\frac{(\ln\tau^2-\mu_{\tau^2})^2}{-2\sigma^2_{\tau^2}}\right)$$

Note that covariance matrix Σ contains the covariance parameters ϕ, σ^2 and τ^2.

C Gibbs Sampling

We show that the full conditional distribution of μ is a normal distribution. By completing the square we have

$$
\begin{aligned}
p(\mu|\phi,\sigma^2,\tau^2,\mathbf{y}) &\propto \exp\left(\frac{(\mathbf{y}-\mu\mathbf{1})^T\Sigma^{-1}(\mathbf{y}-\mu\mathbf{1})}{-2}\right)\exp\left(\frac{(\mu-\mu_\mu)^2}{-2\sigma_\mu^2}\right)\\
&= \exp\left(\frac{-1}{2}(-\mu\mathbf{1}^T\Sigma^{-1}\mathbf{y}+\mu^2\mathbf{1}^T\Sigma^{-1}\mathbf{1}-\mu\mathbf{y}^T\Sigma^{-1}\mathbf{1})-\frac{1}{2}\left(\frac{\mu^2-2\mu_\mu\mu}{\sigma_\mu^2}\right)\right)\\
&= \exp\left(\frac{-1}{2}\left((\mathbf{1}^T\Sigma^{-1}\mathbf{1}+\frac{1}{\sigma_\mu^2})\mu^2-2(\mathbf{1}^T\Sigma^{-1}\mathbf{y}+\frac{\mu_\mu}{\sigma_\mu^2})\mu\right)\right)\\
&= \exp\left(\frac{\mu^2-2\left(\frac{\mathbf{1}^T\Sigma^{-1}\mathbf{y}+\frac{\mu_\mu}{\sigma_\mu^2}}{\mathbf{1}^T\Sigma^{-1}\mathbf{1}+\frac{1}{\sigma_\mu^2}}\right)\mu}{-2\left(\frac{1}{\mathbf{1}^T\Sigma^{-1}\mathbf{1}+\frac{1}{\sigma_\mu^2}}\right)}\right)\\
&= \exp\left(\frac{\left(\mu-\frac{\sigma_\mu^2\mathbf{1}^T\Sigma^{-1}\mathbf{y}+\mu_\mu}{\sigma_\mu^2\mathbf{1}^T\Sigma^{-1}\mathbf{1}+1}\right)^2}{-2\left(\frac{\sigma_\mu^2\mathbf{1}^T\Sigma^{-1}\mathbf{1}+1}{\sigma_\mu^2}\right)^{-1}}\right)
\end{aligned}
$$

which is a normal distribution $N\left(\frac{\sigma_\mu^2\mathbf{1}^T\Sigma^{-1}\mathbf{y}+\mu_\mu}{\sigma_\mu^2\mathbf{1}^T\Sigma^{-1}\mathbf{1}+1},\left(\left(\frac{\sigma_\mu^2\mathbf{1}^T\Sigma^{-1}\mathbf{1}+1}{\sigma_\mu^2}\right)^{-1/2}\right)^2\right)$.

D Checking the Convergence of the MCMC Chains

Figure 8 shows the monitoring plots for μ and ϕ parameters selected at random which are mainly used for determining the MCMC chains length. The plot in the first row is the empirical density function. The second plot is the trend plot which is usually used as a means to check for the convergence to the stationarity distribution. The third plot illustrates the autocorrelation function. As you may see autocorrelation is insignificant for μ; however, the samples of ϕ seems to be autocorrelated. This autocorrelation structure is also verified by looking at the trend plot. High autocorrelation between the samples causes the Effective Sample Size (ESS) to decrease. Note that the "thinning" process eliminates any remaining autocorrelation. The fourth plot shows the variance of the sampled parameters which will become stable (converge to a constant) after reaching the stationarity. The fifth plot demonstrated the mean of the parameter as a function of the sample index. Finally, the last plot shows the standard error of the mean of the parameter which will converge to 0 as the number of samples goes to infinity.

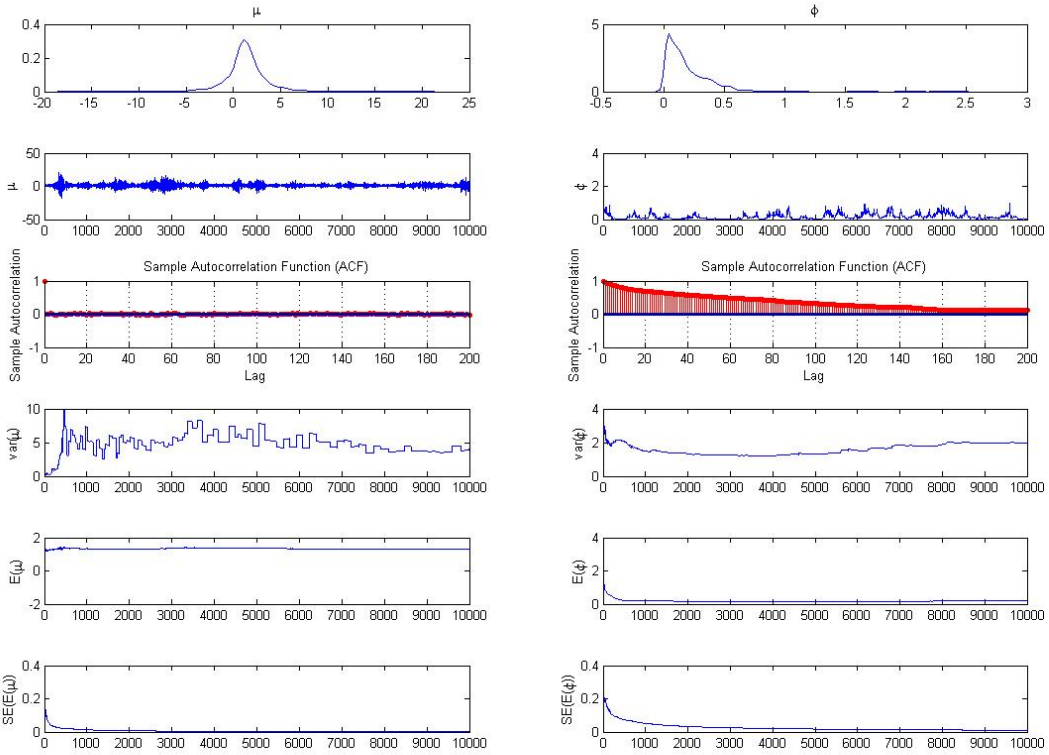


Figure 8: Monitoring plots used to check for convergence of MCMC chains to their stationary distributions - μ (left) and ϕ(right)