EFFICIENT GLOBAL OPTIMIZATION FOR BLACK-BOX SIMULATION VIA SEQUENTIAL INTRINSIC KRIGING

By

Ehsan Mehdad, Jack P.C. Kleijnen

18 August 2015

ISSN 0924-7815
ISSN 2213-9532
Efficient Global Optimization for Black-box Simulation via Sequential Intrinsic Kriging

Ehsan Mehdad and Jack P.C. Kleijnen
Tilburg School of Economics and Management, Tilburg University

July 24, 2015

Abstract

Efficient Global Optimization (EGO) is a popular method that searches sequentially for the global optimum of a simulated system. EGO treats the simulation model as a black-box, and balances local and global searches. In deterministic simulation, EGO uses ordinary Kriging (OK), which is a special case of universal Kriging (UK). In our EGO variant we use intrinsic Kriging (IK), which eliminates the need to estimate the parameters that quantify the trend in UK. In random simulation, EGO uses stochastic Kriging (SK), but we use stochastic IK (SIK). Moreover, in random simulation, EGO needs to select the number of replications per simulated input combination, accounting for the heteroscedastic variances of the simulation outputs. A popular selection method uses optimal computer budget allocation (OCBA), which allocates the available total number of replications over simulated combinations. We derive a new allocation algorithm. We perform several numerical experiments with deterministic simulations and random simulations. These experiments suggest that (1) in deterministic simulations, EGO with IK outperforms classic EGO; (2) in random simulations, EGO with SIK and our allocation rule does not differ significantly from EGO with SK combined with the OCBA allocation rule.

Keywords: Global optimization, Gaussian process, Kriging, intrinsic Kriging, metamodel

JEL: C0, C1, C9, C15, C44

1 Introduction

Optimization methods for black-box simulations—either deterministic or random—have many applications. Black-box simulation means that the input/output (I/O) function is an implicit mathematical function defined by the simulation model (computer code). In many engineering applications of computational fluid dynamics, the computation of the output (response) of a single input combination is usually time-consuming or computationally expensive. In most oper-
ations research (OR) applications, however, a single simulation run is computationally inexpensive, but there are extremely many input combinations; e.g., a single-server queueing model may have one input (namely, the traffic rate) that is continuous, so we can distinguish infinitely many input values but in finite time we can simulate only a fraction of these values. In all these situations, it is common to use metamodels, which are also called emulators or surrogates.

A popular method for the optimization of deterministic simulation is efficient global optimization (EGO), which uses Kriging as the metamodel; see the seminal article Jones et al. (1998). EGO has been adapted for random simulation with either homoscedastic noise variances (see Huang et al. (2006)) or heteroscedastic noise variances (see Picheny et al. (2013) and Quan et al. (2013)). EGO is the topic of much recent research; see Sun et al. (2014), Salemi et al. (2014), and the many references in Kleijnen (2015).

Our first contribution in this paper is the use of intrinsic Kriging (IK) as a metamodel for the optimization of deterministic simulation and random simulation. The basic idea of IK is to remove the trend from I/O data by linear filtration of data. Consequently, IK does not require the second-order stationary condition and it may provide a more accurate fit than Kriging; see Mehdad and Kleijnen (2015).

More specifically, for deterministic simulation we develop EGO with IK as the metamodel. For random simulation we develop stochastic IK (SIK) combined with the two-stage sequential algorithm developed by Quan et al. (2013). The latter algorithm accounts for heteroscedastic noise variances and balances two source of noise; namely, spatial uncertainty due to the metamodel and random variability caused by the simulation. The latter noise is independent from one replication to another; i.e, we suppose that the streams of pseudorandom numbers do not overlap. Moreover we assume that different input combinations do not use common (pseudo)random numbers (CRN), so these input combinations give independent outputs.

Our second contribution concerns Quan et al. (2013)'s two-stage algorithm. We replace the optimal computing budget allocation (OCBA) in the allocation stage of the algorithm by an allocation rule that we build on IK. Our rule allocates the additional replications over the sampled points such that it minimizes the integrated mean squared prediction error (IMSPE).

In our numerical experiments we use test functions of different dimensionality, to study the differences between (1) EGO variants in deterministic simulation; (2) two-stage algorithm variants in random simulation. Our major conclusion will be that in most experiments (1) for deterministic simulations, EGO with IK outperform EGO formulated in Jones et al. (1998); (2) for deterministic simulations, the difference between Quan et al. (2013)'s algorithm and our variant is not significant.

We organize the rest of this paper as follows. Section 2 summarizes classic Kriging. Section 3 explains IK. Section 4 summarizes classic EGO, the two-stage algorithm, and our variant. Section 5 presents numerical experiments. Section 6 summarizes our conclusions.
2 Kriging

In this section we summarize universal Kriging (UK), following Cressie [1991] pp. 151–182). UK assumes

\[ Y(x) = f(x)^T \beta + M(x) \text{ with } x \in \mathbb{R}^d \]  

(1)

where \( Y(x) \) is a random process at the point (or input combination) \( x \), \( f(x) \) is a vector of \( p + 1 \) known regression functions or trend, \( \beta \) is a vector of \( p + 1 \) parameters, and \( M(x) \) is a stationary GP with zero mean and covariance function \( \Sigma_M \).

This \( \Sigma_M \) must be specified such that it makes \( M(x) \) in (1) a stationary GP; i.e., \( \Sigma_M \) is a function of the distance between the points \( x_i \) and \( x_{i'} \) with \( i, i' = 0, 1, \ldots, m \) where the subscript 0 denotes a new point and \( m \) denotes the number of old points. Separable anisotropic covariance functions use the distances along the \( d \) axes \( h_{i;i'g} = |x_{ig} - x_{i'g}| \) \( (g = 1, \ldots, d) \). The most popular choice for such a function is the so-called Gaussian covariance function:

\[ \text{cov}(x_i, x_{i'}) = \sigma^2 \prod_{g=1}^d \exp\left(-\theta_g h_{i;i'g}^2\right) \text{ with } \theta_g > 0 \]  

(2)

where \( \sigma^2 \) is the variance of \( M(x) \).

Let \( Y = (Y(x_1), \ldots, Y(x_m))^\top \) denote the vector with the \( m \) values of the metamodel in (1) at the \( m \) old points. Kriging predicts \( Y \) at a (new or old) point \( x_0 \) linearly from the old I/O data \( (X, Y) \) where \( X = (x_1, \ldots, x_m) \) is the \( d \times m \) matrix with \( m \) points \( x_i = (x_{g,i}) \) \( (i = 1, \ldots, m; \ g = 1, \ldots, d) \):

\[ \hat{Y}(x_0) = \lambda^\top Y \text{ such that } \lambda^\top F = f(x_0)^\top \]  

(3)

where \( F \) is the \( m \times (p + 1) \) matrix with element \( (i, j) \) being \( f_j(x_i) \), \( f(x_0) = (f_0(x_0), \ldots, f_p(x_0)) \), and the condition for \( \lambda \) guarantees that \( \hat{Y}(x_0) \) is an unbiased predictor. The optimal linear unbiased predictor minimizes the mean squared prediction error (MSPE), defined as

\[ \text{MSPE}(\hat{Y}(x_0)) = \mathbb{E}(\hat{Y}(x_0) - Y(x_0))^2. \]

Cressie [1991] pp. 151–157) shows how to use Lagrangian multipliers to solve this constrained minimization problem, which gives the optimal weights and the predictor:

\[ \lambda^\top = \left( \Sigma_M(x_0, \cdot) + F (F^\top \Sigma_M^{-1} F)^{-1} (f(x_0) - F^\top \Sigma_M^{-1} \Sigma_M(x_0, \cdot)) \right)^\top \Sigma_M^{-1} \]

\[ \hat{Y}(x_0) = \lambda^\top Y \]  

(4)

with \( \Sigma_M(x_0, \cdot) = (\Sigma_M(x_0, x_1), \ldots, \Sigma_M(x_0, x_m))^\top \) denoting the \( m \)-dimensional vector with covariances between the outputs of the one new point and the \( m \) old points, and \( \Sigma_M \) denoting the \( m \times m \) matrix with the covariances between the
old outputs, so the element \((i,i')\) is \(\Sigma_M(x_i,x_{i'})\). The resulting minimal MSPE is

\[
\text{MSPE}(\hat{Y}(x_0)) = \tau^2 - \Sigma_M(x_0,\cdot)\Sigma_M^{-1}(x_0,\cdot) + (f(x_0) - F^\top \Sigma_M^{-1}(x_0,\cdot)) \left( F^\top \Sigma_M F \right)^{-1} (f(x_0) - F^\top \Sigma_M^{-1}(x_0,\cdot)).
\]

(5)

This Kriging is an exact interpolator; i.e., for the old points, \(\hat{Y}(x_i)\) gives a predictor that equals the observed output. Important for EGO is the property that for the old points the Kriging variance reduces to zero.

The Kriging metamodel defined in \((1)\) can be extended to incorporate the so-called internal noise in random simulation; see Opsomer et al. \(\{1999\}\); Ginsbourger \(\{2009\}\); Ankenman et al. \(\{2010\}\); Yin et al. \(\{2011\}\). The resulting stochastic Kriging (SK) metamodel at replication \(r\) of the random simulation output at \(x\) is

\[
Z_r(x) = f(x)^\top \beta + M(x) + \varepsilon_r(x) \quad \text{with} \quad x \in \mathbb{R}^d
\]

where \(\varepsilon_1(x), \varepsilon_2(x), \ldots\) denotes the internal noise at point \(x\). We assume that \(\varepsilon_r(x)\) has a Gaussian distribution with mean zero and variance \(V(x)\) and that it is independent of \(M(x)\).

We assumed that the external noise \(M(x)\) and the internal noise \(\varepsilon(x)\) in \((6)\) are independent, so the SK predictor and its MSPE can be derived analogously to the derivation for UK in \((4)\) and \((5)\) except that \(\Sigma = \Sigma_M + \Sigma_r\) replaces \(\Sigma_M\) where \(\Sigma_r\) is a diagonal matrix \(\text{(no CRN)}\) with the variances of the internal noise \(V(x_i)/n_i\) on the main diagonal, and \(\Sigma_M\) still denoting the covariance matrix of Kriging without internal noise. We also replace \(Y\) in \((4)\) and \((5)\) by the sample mean \(\bar{Z} = (\bar{Z}(x_1), \ldots, \bar{Z}(x_m))^\top\).

### 3 Intrinsic Kriging

In this section we explain IK, following Mehdad and Kleijnen \(\{2015\}\). We rewrite \((1)\) as

\[
\mathbf{Y} = \mathbf{F} \beta + \mathbf{M}
\]

(7)

with \(\mathbf{Y} = (\mathbf{Y}(x_1), \ldots, \mathbf{Y}(x_m))^\top\) and \(\mathbf{M} = (\mathbf{M}(x_1), \ldots, \mathbf{M}(x_m))^\top\). We do not assume \(\mathbf{M}\) is second-order stationary. Let \(Q\) be an \(m \times m\) matrix such that \(Q\mathbf{F} = \mathbf{O}\) where \(\mathbf{O}\) is an \(m \times (p+1)\) matrix with all elements zero. Together, \(Q\) and \((7)\) give

\[
Q\mathbf{Y} = Q\mathbf{M}.
\]

Consequently, the second-order properties of \(Q\mathbf{Y}\) depend on \(Q\mathbf{M}\) and not on the regression function \(\mathbf{F}\beta\).

To generalize the metamodel in \((1)\), we need a stochastic process for which \(Q\mathbf{M}\) is second-order stationary; such processes are called intrinsically stationary processes. We assume that \(f_j(x) (j = 1, \ldots, p+1)\) are mixed monomials \(x_1^{i_1} \cdots x_d^{i_d}\) with \(x = (x_1, \ldots, x_d)^\top\) and nonnegative integers \(i_1, \ldots, i_d\) such that \(i_1 + \ldots + i_d \leq k\) with \(k\) a given nonnegative integer. An intrinsic random function of order \(k\) (IRF-\(k\)) is a random process \(\mathbf{Y}\) for which \(\sum_{i=1}^m \lambda_i \mathbf{Y}(x_i)\) with
\( \mathbf{x}, \in \mathbb{R}^d \) is second-order stationary, and \( \mathbf{\lambda} = (\lambda_1, \ldots, \lambda_m)^\top \) is a generalized-increment vector of real numbers such that
\[
(f_j(\mathbf{x}_1), \ldots, f_j(\mathbf{x}_m)) \mathbf{\lambda} = 0 \ (j = 1, \ldots, p + 1).
\]

IK is based on an IRF-\( k \). Let \( \mathcal{M}(\mathbf{x}) \) be an IRF-\( k \) with mean zero and generalized covariance function \( K \). Then the corresponding IK metamodel is:
\[
\mathcal{Y}(\mathbf{x}) = \mathbf{f}(\mathbf{x})^\top \mathbf{\beta} + \mathcal{M}(\mathbf{x}). \tag{8}
\]

[\text{Cressie} \ 1991, pp. 299–306] derives a linear predictor for the IRF-\( k \) metamodel defined in (8) with generalized covariance function \( K \). So—given the old outputs \( \mathcal{Y} = (\mathcal{Y}(\mathbf{x}_1), \ldots, \mathcal{Y}(\mathbf{x}_m))^\top \)—the optimal linear predictor \( \hat{\mathcal{Y}}(\mathbf{x}_0) = \mathbf{\lambda}^\top \mathcal{Y} \) follows from minimizing the MSPE of the linear predictor:
\[
\min_{\mathbf{\lambda}} E \left( (\hat{\mathcal{Y}}(\mathbf{x}_0) - \mathcal{Y}(\mathbf{x}_0))^2 \right).
\]

IK should meet the condition
\[
\mathbf{\lambda}^\top \mathbf{F} = (f_0(\mathbf{x}_0), \ldots, f_p(\mathbf{x}_0)). \tag{9}
\]

This condition is not introduced as the unbiasedness condition; instead the condition guarantees that the coefficients of the prediction error \( \mathcal{Y}_1 \mathcal{Y}(\mathbf{x}_1) + \ldots + \mathcal{Y}_m \mathcal{Y}(\mathbf{x}_m) - \mathcal{Y}(\mathbf{x}_0) \) create a generalized-increment vector \( \mathbf{\lambda}_{m+1} = (\mathbf{\lambda}^\top, \lambda_0) \) with \( \lambda_0 = -1 \). The IK variance, denoted by \( \sigma_{\text{IK}}^2 \) is:
\[
\sigma_{\text{IK}}^2 = \text{var}(\mathbf{\lambda}_{m+1}^\top \mathcal{Y}) = \sum_{i=0}^m \sum_{i'=0}^m \lambda_i \lambda_{i'} K(\mathbf{x}_i, \mathbf{x}_{i'}). \tag{10}
\]

In this section we assume that \( K \) is known, so the optimal linear predictor is obtained through minimization of (10) subject to (9). Hence, the IK predictor is
\[
\hat{\mathcal{Y}}(\mathbf{x}_0) = \mathbf{\lambda}^\top \mathcal{Y}, \tag{11}
\]
\[
\mathbf{\lambda}^\top = \left( \mathbf{K}(\mathbf{x}_0, \cdot) + \mathbf{F}(\mathbf{F}^\top \mathbf{K}^{-1} \mathbf{F})^{-1} (\mathbf{f}(\mathbf{x}_0) - \mathbf{F}^\top \mathbf{K}^{-1} \mathbf{K}(\mathbf{x}_0, \cdot)) \right)^\top \mathbf{K}^{-1}
\]

where \( \mathbf{K}(\mathbf{x}_0, \cdot) = (K(\mathbf{x}_0, \mathbf{x}_1), \ldots, K(\mathbf{x}_0, \mathbf{x}_m))^\top \) and \( \mathbf{K} \) is a \( m \times m \) matrix with the \( (i, i') \) element \( K(\mathbf{x}_i, \mathbf{x}_{i'}) \). The resulting \( \sigma_{\text{IK}}^2 \) is
\[
\text{MSPE}(\hat{\mathcal{Y}}(\mathbf{x}_0)) = \mathbf{K}(\mathbf{x}_0, \mathbf{x}_0) - \mathbf{K}(\mathbf{x}_0, \cdot)^\top \mathbf{K}^{-1} \mathbf{K}(\mathbf{x}_0, \cdot) + \\
(f(\mathbf{x}_0) - \mathbf{F}^\top \mathbf{K}^{-1} \mathbf{K}(\mathbf{x}_0, \cdot))^\top (\mathbf{F}^\top \mathbf{K}^{-1} \mathbf{F})^{-1} (f(\mathbf{x}_0) - \mathbf{F}^\top \mathbf{K}^{-1} \mathbf{K}(\mathbf{x}_0, \cdot)) \tag{12}
\]

Now we discuss properties of \( K \). Obviously \( K \) is symmetric. Moreover \( K \) must be conditionally positive definite so
\[
\text{var}(\mathbf{\lambda}^\top \mathcal{Y}) = \sum_{i=1}^m \sum_{i'=1}^m \lambda_i \lambda_{i'} K(\mathbf{x}_i - \mathbf{x}_{i'}) \geq 0, \ \forall \mathbf{\lambda} \text{ such that } (f_j(\mathbf{x}_1), \ldots, f_j(\mathbf{x}_m)) \mathbf{\lambda} = 0 \]
where the condition must hold for $j = 1, \ldots, p + 1$. Following Mehdad and Kleijnen (2015) we choose $K$ as

$$K(x, x'; \theta) = \prod_{g=1}^{d} \left( \theta_{0;g} + \theta_{1;g} \int_{0}^{1} \frac{(x_g - u_g)^{k_g} (x'_g - u_g)^{k_g}}{(k_g!)^2} du_g \right)$$  \quad (13)

where $\theta = (\theta_{0;1}, \theta_{1;1}, \theta_{0;2}, \ldots, \theta_{0;d}, \theta_{1;d}) \geq 0$. The function in (13) accepts different $k$ for different input dimensions, so we have a vector of the orders $k = (k_1, \ldots, k_d)^\top$.

In practice, $K$ is unknown so we estimate the parameters $\theta$. For this estimation we use restricted maximum likelihood (REML) estimator, which maximizes the likelihood of transformed data that do not contain the unknown parameters of the trend. This transformation is close to the concept of intrinsic Kriging. So we assume that $\mathcal{Y}$ in $\mathcal{Y}(x)$ is a Gaussian IRF-$k$. The REML estimator of $\theta$ is then found through minimization of the negative log-likelihood function

$$\ell(\theta) = (m - q)/2 \log(2\pi) - \frac{1}{2} \log |\mathbf{F}^\top \mathbf{F}| + \frac{1}{2} \log |K(\theta)| + \frac{1}{2} \log |\mathbf{F}^\top K(\theta)^{-1} \mathbf{F}|$$

$$+ \frac{1}{2} \mathcal{Y}^\top \mathbf{K}(\theta) \mathcal{Y}$$  \quad (14)

where $q = \text{rank}(\mathbf{F})$ and $\mathbf{K}(\theta) = K(\theta)^{-1} - K(\theta)^{-1} \mathbf{F} (\mathbf{F}^\top K(\theta)^{-1} \mathbf{F})^{-1} \mathbf{F}^\top K(\theta)^{-1}$. Finally, we replace $K$ by $K(\hat{\theta})$—in (11) to obtain $\mathcal{Y}(x_0)$ and in (12) to obtain $\hat{\sigma}_y^2$. We could require REML to estimate the optimal (integer) $k^*$, but this requirement would make the optimization even more difficult; i.e., we set $k = 0$.

Mehdad and Kleijnen (2015) also extends IK to account for random simulation output with noise variances that change across the input space. The methodology is similar to the extension of Kriging to stochastic Kriging. The interpolating property of IK does not make sense for random simulation, which has sampling variability or internal noise besides the external noise or spatial uncertainty created by the fitted metamodel.

The extension of IK to account for internal noise with a constant noise variance has already been studied in the (geostatistics) literature as a so-called nugget effect. Indeed, Cressie (1991, p. 305) replaces $K$ by $K + c_0 \delta(h)$ where $c_0 \geq 0$, $\delta(h) = 0$ if $h > 0$, and $\delta(h) = 1$ if $h = 0$. Mehdad and Kleijnen (2015) considers the case of heteroscedastic noise variances similar to SK, the stochastic IK (SIK) metamodel at replication $r$ of the random output at $x$ is

$$\mathcal{Y}_r(x) = \mathbf{f}(x)^\top \beta + \mathcal{M}(x) + \varepsilon_r(x) \text{ with } x \in \mathbb{R}^d$$  \quad (15)

where $\varepsilon_1(x), \varepsilon_2(x), \ldots$ denotes the internal noise at $x$. We again assume that $\varepsilon(x)$ has a Gaussian distribution with mean zero and variance $V(x)$ and that $\varepsilon(x)$ is independent of $M(x)$.

In stochastic simulation, the experimental design consists of pairs $(x_i, n_i)$, $i = 1, \ldots, m$, where $n_i$ is the number of replications at $x_i$. These replications enable computing the classic unbiased estimators of the mean output and the internal
we choose the number of replications at each old point \( n \) in (11) and (12) by the generalized covariance matrix of \( \mathbf{K} \) without internal noise. We also replace \( \mathbf{V} \) in (11) and (12)—except that \( \mathbf{K}_m \) and \( \mathbf{K}_r \) are diagonal matrices with \( \mathbf{V}(\mathbf{x}_i)/n_i \) on the main diagonal, and \( \mathbf{K}_m \) still denoting the generalized covariance matrix of \( \mathbf{K} \) without internal noise. We also replace \( \mathbf{V} \) in (11) and (12) by \( \mathbf{Y} = (\nabla(\mathbf{x}_1), \ldots, \nabla(\mathbf{x}_m))^\top \). So the SIK predictor is

\[
\hat{\mathbf{Y}}(\mathbf{x}_0) = \mathbf{X}^\top \mathbf{Y}
\]

and its MSPE is

\[
\text{MSPE}(\hat{\mathbf{Y}}(\mathbf{x}_0)) = \mathbf{K}_m(\mathbf{x}_0, \mathbf{x}_0) - \mathbf{K}_m(\mathbf{x}_0, \cdot)^\top \mathbf{K}^{-1} \mathbf{K}_m(\mathbf{x}_0, \cdot) + \\
(\mathbf{f}(\mathbf{x}_0) - \mathbf{f}^\top \mathbf{K}^{-1} \mathbf{K}_m(\mathbf{x}_0, \cdot))\top (\mathbf{f}^\top \mathbf{K}^{-1} \mathbf{K}_m(\mathbf{x}_0, \cdot)) - \\
(\mathbf{f}(\mathbf{x}_0) - \mathbf{f}^\top \mathbf{K}^{-1} \mathbf{K}_m(\mathbf{x}_0, \cdot))\top (\mathbf{f}^\top \mathbf{K}^{-1} \mathbf{K}_m(\mathbf{x}_0, \cdot)).
\]

We again use REML to estimate the parameters \( \theta \) of the generalized covariance function, and replace \( \mathbf{K}_m \) by \( \mathbf{K}_m(\hat{\theta}) \). We also need to estimate the internal noise \( \mathbf{V} \) which is typically unknown. Let \( \hat{\mathbf{V}}(\mathbf{x}_i) = s^2(\mathbf{x}_i) \) be the estimator of \( \mathbf{V}(\mathbf{x}_i) \), so we replace \( \mathbf{K}_r \) by \( \mathbf{K}_r = \left( \hat{\mathbf{V}}(\mathbf{x}_1)/n_1, \ldots, \hat{\mathbf{V}}(\mathbf{x}_m)/n_m \right) \). Finally, we replace \( \mathbf{K} = \mathbf{K}_m + \mathbf{K}_r \) by \( \hat{\mathbf{K}} = \mathbf{K}_m(\hat{\theta}) + \mathbf{K}_r \) in (17) and (18). Next we explain how we choose the number of replications at each old point \( n_i \).

We are interested in an experimental design that gives a low integrated MSPE (IMSP). Following Mehdad and Kleijnen (2015)—who revised Ankenman et al. (2010)—we allocate \( N \) replications among \( m \) old points \( \mathbf{x}_i \) such that this design minimizes the IMSP. Let \( \mathcal{X} \) be the design space. Then the design is meant to give

\[
\min_n \text{IMSP}(\mathbf{x}_i) = \min_n \int_{\mathbf{x}_0 \in \mathcal{X}} \text{MSPE}(\mathbf{x}_0, n) d\mathbf{x}_0
\]

subject to \( n^\top \mathbf{1}_m \leq N \), and \( \mathbf{n} = (n_1, \ldots, n_m)^\top \) where \( n_i \in \mathbb{N} \).

We try to compute \( n_i^* \), which denotes the optimal allocation of the total number of replications \( N \) over the \( m \) old points. We then run into the problem that \( n_i^* \) is determined by both \( \mathbf{K}_m \) (external noise) and \( \mathbf{K}_r \) (internal noise). To solve this problem, we follow Ankenman et al. (2010) and ignore \( \mathbf{K}_r \) so \( \mathbf{K} \approx \mathbf{K}_m \), and we relax the integrality condition. Altogether we derive

\[
n_i^* \approx N \frac{\sqrt{\mathbf{V}(\mathbf{x}_i)} C_i}{\sum_{i=1}^m \sqrt{\mathbf{V}(\mathbf{x}_i)} C_i}, \quad \text{with } C_i = [\mathbf{S}^{-1} \mathbf{W}^{-1}]_{p+1+i, p+1+i}
\]

subject to \( n^\top \mathbf{1}_m \leq N \), and \( \mathbf{n} = (n_1, \ldots, n_m)^\top \) where \( n_i \in \mathbb{N} \).
where
\[
S = \begin{bmatrix} O & F^T \\ F & K \end{bmatrix}, \quad W = \int \begin{bmatrix} f(x_0)f(x_0)^T & f(x_0)K_M(x_0, \cdot)^T \\ K_M(x_0, \cdot)f(x_0)^T & K_M(x_0, \cdot)K_M(x_0, \cdot)^T \end{bmatrix} \, dx_0,
\]
and \(J^{(i)}\) is a \((p+1+m) \times (p+1+m)\) matrix with 1 in position \((p+1+i, p+1+i)\) and zeros elsewhere. Note that in (19) both the internal noise variance \(V(x)\) and the external noise covariance function \(K_M\) affect the allocation.

4 Efficient global optimization (EGO)

In this section we first summarize EGO and three variants; next we detail one variant, which is the focus of our article.

4.1 EGO and three variants

EGO is the global optimization algorithm developed by [Jones et al. 1998] for deterministic simulation. It uses expected improvement (EI) as its criterion to balance local and global search or exploiting and exploring. EGO is a sequential method with the following steps.

1. Fit a Kriging metamodel to the old I/O data. Let \(f_{\min} = \min_i Y(x_i)\) be the minimum function value observed (simulated) so far.

2. Estimate \(x_0\) that maximizes \(EI(x) = \mathbb{E}[\max(f_{\min} - Y_p(x), 0)]\). Assuming \(Y_p(x) \sim \mathcal{N}(\hat{Y}(x), \hat{\sigma}^2(x))\), [Jones et al. 1998] derive
\[
\hat{EI}(x) = \left(f_{\min} - \hat{Y}(x)\right) \Phi \left(\frac{f_{\min} - \hat{Y}(x)}{\hat{\sigma}(x)}\right) + \hat{\sigma}(x) \phi \left(\frac{f_{\min} - \hat{Y}(x)}{\hat{\sigma}(x)}\right)
\]
where \(\hat{Y}(x)\) is defined in (4) and \(\hat{\sigma}^2(x)\) follows from (5) substituting estimators for \(\tau^2\) and \(\theta\); \(\Phi\) and \(\phi\) denote the cumulative distribution function (CDF) and probability density function (PDF) of the standard normal distribution.

3. Simulate the response at \(\hat{x}_0\) found in step 2. Fit a new Kriging metamodel to the old and new points. Return to step 1, unless the \(\hat{EI}\) satisfies a given criterion; e.g., \(\hat{EI}\) is less than 1% of the current best function value.

[Huang et al. 2006] adapts EI for random simulation. That article uses the metamodel defined in (15) and assumes that the noise variances are identical across the design space; i.e., \(V(x) = \hat{V}\). That article introduces the following augmented EI (AEI) to balance exploitation and exploration:
\[
\hat{AEI}(x) = \mathbb{E}\left[\max\left(\hat{Z}(x^*) - Z_p(x), 0\right)\right] \left[1 - \left(\frac{\hat{V}}{\text{MSPE}(\hat{Z}(x)) + \hat{V}}\right)^{1/2}\right]
\]

8
where \( x^* \) is called the current ‘effective best solution’ and is defined as \( x^* = \arg \min_{x_1, \ldots, x_m} [\hat{Z}(x) + \text{MSPE}(\hat{Z}(x))] \); the second term on the right-hand side in (21) accounts for the diminishing returns of additional replications at the current best point.

Picheny et al. (2013) develops a quantile-based EI known as \textit{expected quantile improvement} (EQI). This criterion lets the user specify the risk level; i.e., the higher the values for the quantile are specified, the more conservative the criterion becomes. The EQI accounts for a limited computation budget; moreover, to sample a new point, EQI also considers the noise variance at future (not yet sampled) points. However, EQI requires a known variance function for the noise, and Quan et al. (2013) claims that Picheny et al. (2013)’s algorithm is also computationally more complex.

Quan et al. (2013) shows that EI and AEI can not be good criteria for random simulations with heteroscedastic noise variances. The argument is that an EGO-type framework for random simulation with heteroscedastic noise faces three challenges: (1) An effective procedure should locate the global optimum with a limited computing budget. (2) To balance exploration and exploitation in random simulation, a new procedure should be able to search globally without exhaustively searching a local region; a good estimator of \( f_{\min} \) is necessary especially when there are several points that give outputs close to the global optimum. (3) With a limited computing budget, it is wise to obtain simulation observations in unexplored regions in the beginning of the search and—as the budget is being expended toward the end—focus on improving the current best area. Quan et al. (2013) finds no significant difference between its new two-stage algorithm and Picheny et al. (2013)’s algorithm.

4.2 Quan et al. (2013)’s algorithm

Now we explain Quan et al. (2013)’s algorithm in detail. After the initial fit of a SK metamodel, each iteration of the algorithm consists of a search stage followed by an allocation stage. In the search stage, the \textit{modified expected improvement} (MEI) criterion is used to select a new point; see (22) below. Next in the allocation stage, OCBA distributes an additional number of replications over the sampled points. Altogether, the search stage locates the potential global minima, and the allocation stage reduces the noise caused by random variability at sampled points to improve the metamodel in regions where local minima exist and finally selects the global optimum.

Quan et al. (2013)’s algorithm has a ‘division of allocation heuristic’. The computing budget per iteration is set as a constant, but the allocation of this budget between the searching stage and the allocation stage changes as the algorithm progresses. In the beginning, most of the budget is invested in exploration (search stage). During the progress of the algorithm, the focus moves to identifying the point with the lowest sample mean (allocation stage).

In the search stage, we compute

\[
\text{MEI}(x) = \mathbb{E}[\max(\hat{Z}_{\min} - Z_P(x), 0)]
\] (22)
where $\hat{Z}_{\text{min}}$ is the predicted response at the sampled point with the lowest sample mean, and $Z_p$ is a normal random variable with the mean equal to $\hat{Z}(x)$ and the variance equal to the estimated MSPE($\hat{Z}(x)$). The allocation stage addresses the random noise, so the search stage uses only MSPE($\hat{Z}(x)$) with estimates of $\Sigma = \Sigma_M$ instead of $\Sigma = \Sigma_M + \Sigma_r$. This helps the search to focus on the new points that reduce the spatial uncertainty of the metamodel. Ignoring the uncertainty caused by random variability, the MEI criterion assumes that the observations are made with infinite precision so the same point is never selected again. This helps the algorithm to quickly escape from a local optimum, and makes the sampling behavior resembles the behavior of the original EI criterion with its balancing of exploration and exploitation.

The allocation stage reduces random variability through the allocation of additional replications among sampled points. Additional replications are distributed with the goal of maximizing the probability of the correct selection (PCS) when selecting a sampled point as the global optimum. Suppose that we have $m$ sampled points $x_i$ ($i = 1, \ldots, m$) with sample mean $Z_i$ and sample variance $\hat{V}(x_i) = s^2(x_i)$. Then the approximate PCS (APCS) can be asymptotically maximized when the available computing budget $N$ tends to infinity and

$$
\begin{align*}
\frac{n_i}{n_j} &= \left( \frac{\hat{V}(x_i)}{\Delta_{b,i}} \right)^2, \\
n_b &= \frac{\hat{V}(x_b)}{\sqrt{\sum_{i=1, i \neq b}^m n_i^2 \hat{V}(x_i)}},
\end{align*}
$$

(23) (24)

where $n_i$ is the number of replications allocated to $x_i$, $x_b$ is the point with the lowest sample mean and the sample mean at point $x_i$; Chen et al. (2000) proves (23) and (24). Given this allocation rule, at the end of the allocation stage, the sampled point with the lowest sample mean will be selected as $\hat{Z}_{\text{min}}$.

We summarize Quan et al. (2013)'s algorithm in our Algorithm 1. Before the algorithm begins, the user must specify $T$, $B$, $m_0$, and $r_{\text{min}}$ where $T$ is the total number of replications at the start, $B$ is the number of replications available for each iteration, $m_0$ is the size of the initial space filling design, and $r_{\text{min}}$ is the minimum number of replications for a new point. The size of the initial design $m_0$ may be set to $10^d$ where $d$ is the number of dimensions ($10^d$ is proposed by Loeppky et al. (2009)); $B$ and $r_{\text{min}}$ should be set such that there are sufficient replications available for the first allocation stage.

The starting parameters that determine the number of iterations, $I = \lceil(T - m_0B)/B\rceil$, and the computing budget used per iteration $B$ are set prior to collecting any data, so the starting parameter settings may turn out to be unsuitable for the problem. In step 2, leave-one-out cross validation can provide feedback regarding the suitability of the initial parameters. If one or more design points fail the cross-validation test, then the computing budget may be insufficient to deal with the internal noise. Possible solutions include (i) increasing
Algorithm 1 Quan et al. (2013)’s algorithm

Step 1. Initialization: Run a space filling design with \( m_0 \) points, with \( B \) replications allocated to each point.

Step 2. Validation: Fit a SK metamodel to the set of sample means. Use leave-one-out cross validation to check the quality of the initial SK.

Step 3. Set \( i = 1, r_A(0) = 0 \)

while \( i \leq I \) do

\[ r_A(i) = r_A(i-1) + \min \left( \left\lfloor \frac{B-r_{min}}{I} \right\rfloor, T-m_0B-(i-1)B \right) \]

if \((T-m_0B-(i-1)B-r_A(i))>0\) then

\[ r_S(i) = B - r_A(i) \]

Step 3a. Search Stage: Sample a new point that maximizes the MEI criterion (Eq. 22) with \( r_S(i) \) replications.

Step 3b. Allocation Stage: Using OCBA (Eq. 23 and Eq. 24), allocate \( r_A(i) \) replications among all sampled points.

Step 3c. Fit a SK metamodel to the set of sample means

\( i = i + 1 \)

end if

end while

The point with the lowest sample mean at the end estimates the global optimum.

---

5 Numerical experiments

We experiment with deterministic simulation and random simulation. In these experiments we use a zero-degree polynomial for the trend (so \( p = 0 \) in [1]), so UK becomes ordinary Kriging (OK). For deterministic simulation we study the performance of EGO with OK versus EGO with IK. For random simulation we study the performance of Quan et al. (2013)’s algorithm versus our variant with SIK (instead of SK) and the minimum IMSE allocation rule (instead of OCBA).
We tried to use the MATLAB code developed by Yin et al. (2011)—which is a building block in Quan et al. (2013)’s algorithm—to experiment with the Kriging variants (OK for deterministic simulation and SK for random simulation), but that MATLAB code crashed in experiments with $d > 1$. We therefore use the R package DiceKriging to implement OK and SK; for details on DiceKriging we refer to Roustant et al. (2012). We implement our code for IK and SIK in MATLAB, as Mehdad and Kleijnen (2015) also does. In all our experiments we select $k = 0$.

Furthermore, we select a set of $m_c = 100d$ candidate points, and as the ‘winning’ point we pick the candidate point that maximizes EI or MEI. For $d = 1$ we select $m_0$ and $m_c$ equispaced points; for $d > 1$ we select select $m_0$ and $m_c$ space-filling points through Latin hypercube sampling (LHS), implemented in the MATLAB function lhsdesign.

As the criterion for comparing the performance of different optimization algorithms, we use the number of simulated input combinations needed to estimate the optimal input combination (say) $m$. As the stopping criterion we select $m$ reaching a limit; namely, 11 for $d = 1$, 61 for the camel-back test function ($d = 2$), 65 for Hartmann-3 ($d = 3$), and 111 for Ackley-5 ($d = 5$). We select the number of starting points $m_0$ to be 3 for $d = 1$, 21 for $d = 2$, 30 for $d = 3$, and 51 for $d = 5$.

### 5.1 Deterministic simulation experiments

We experiment with several multi-modal test functions of different dimensionality. We start with Gramacy and Lee (2012)’s test function with $d = 1$:

$$f(x) = \frac{\sin(10\pi x)}{2x} + (x - 1)^4 \text{ and } 0.5 < x \leq 2.5.$$  \hspace{1cm} (25)

Next we experiment with several functions that are popular in optimization; see Dixon and Szego (1978) and http://www.sfu.ca/~ssurjano/index.html:

1. Six-hump camel-back with $d = 2$
2. Hartmann-3 with $d = 3$
3. Ackley-5 with $d = 5$; we define these three test functions in the appendix.

Figure 1 illustrates EGO with IK for the $d = 1$ test function defined in (25). This function has a global minimum at $x_{opt} = 0.5486$ with output $f(x_{opt}) = -0.869$; also see the curves in the left panels of the figure, where the (blue) solid curve is the true function and the (red) dotted line is the IK metamodel. We start with $m = 3$ old points, and stop after sequentially adding seven new points (shown by black circles); i.e., from top to bottom the number of points increases starting with three points and ending with ten points. These plots show that a small $m$ gives a poor metamodel. The right panels display $\hat{EI}$ as $m$ increases. These panels show that $\hat{EI} = 0$ at the old points.

Figure 2 displays $f_{min}(m) = \min f(\hat{x}_i) \ (1 \leq i \leq m)$, which denotes the estimated optimal simulation output after $m$ simulated input combinations; horizontal lines mean that the most recent simulated point does not give a lower estimated optimal output than a preceding point. The square marker represents EGO with IK and the circle marker represents classic EGO with
Figure 1: EGO with IK for Gramacy and Lee’s test function

(a) EI with IK and OK in Gramacy
(b) EI with IK and OK in camel-back
(c) EI with IK and OK in Hartmann-3
(d) EI with IK and OK in Ackley-5

Figure 2: Estimated optimal output (y-axis) after m simulated input combinations (x-axis) in four test functions, for deterministic simulation
OK. The results show that in most experiments, EGO with IK performs better than classic EGO with OK; i.e., EGO with IK gives a better input combination after fewer simulated points.

5.2 Random simulation experiments

In this subsection we compare the performance of Quan et al. (2013)'s algorithm with our variant which uses SIK as the metamodel and a different allocation rule. In both variants of the algorithm we select $r_{\text{min}} = 10$, $B = 40$ (for $d = 1$), 130 (for $d = 2$ and 3), 310 (for $d = 5$). In all our experiments for random simulation we augment the deterministic test functions defined for our deterministic experiments with the heteroscedastic noise $V(x_i) = (1 + |y(x_i)|)^2$.

Figure 3 illustrates our variant for the $d = 1$ test function. We start with $m = 3$ old points, and stop after sequentially adding seven new points. With small $m$ and high noise (as $x$ increases), the metamodel turns out to be a 'poor' approximation; compare the (blue) solid curves and the (red) dashed curves. Note that in the beginning the algorithm searches the region far from the unknown global optimum (namely, $x = 0.5486$), and after each iteration and careful allocation of added replications, the quality of the SIK fit in areas with high noise (when moving to the right) improves. The right panels display $\hat{\text{M}}E\text{I}$ as $m$ increases.

Finally, we compare the two variants using 50 macro-replications. Figure 4 displays $f_{\text{min}}(m) = \min_i \left( \sum_{t=1}^{50} f_t(\hat{x}_i) / 50 \right)$, $1 \leq i \leq m$; which denotes the estimated optimal simulation output averaged over 50 macro-replications. The circle marker represents Quan et al. (2013)'s original algorithm, and the square marker represents our variant. The results show that the two variants are not significantly different in most sampled points except for Hartmann-3 where the original variant performs significantly better for $m = 32, \ldots, 45$ and our variant perform significantly better for $m = 31, 59, \ldots, 65$. We note that this conclusion is confirmed by paired t-tests, which we do not detail here.

6 Conclusions

We adapted EGO replacing OK by IK for deterministic simulation, and for stochastic simulation we modified Quan et al. (2013)’s algorithm replacing SK by a SIK metamodel and introducing a new allocation rule. We numerically compared the performance through numerical experiments with several classic test functions. Our main conclusion is that in most experiments; (i) in deterministic simulations, EGO with IK performs better than Jones et al. (1998)’s EGO with OK; (ii) in random simulation, there is no significant difference between the two variants of Quan et al. (2013)’s algorithm.

In future research we may further investigate the allocation of replications in random simulation analyzed by Kriging metamodels. Furthermore, we would like to see more practical applications of our methodology.
Figure 3: Our variant of Quan et al. (2013)’s algorithm for Gramacy and Lee (2012)’s function

Figure 4: Estimated optimal output (y-axis) after m simulated input combinations (x-axis) for four test functions, for random simulation
A Test functions with dimensionality \( d > 1 \)

We define the following three types of test functions.

1. Six-hump camel-back with \(-2 \leq x_1 \leq 2, -1 \leq x_2 \leq 1, x_{\text{opt}}^\top = (\pm 0.0898, \mp 0.7126)\), and \( f(x_{\text{opt}}) = -1.0316 \)

\[
f(x_1, x_2) = 4x_1^2 - 2.1x_1^4 + x_1^6/3 + x_1x_2 - 4x_2^2 + 4x_2^4.\]

2. Hartmann-3 function with \( 0 \leq x_i \leq 1, i = 1, 2, 3, x_{\text{opt}}^\top = (0.114614, 0.555649, 0.852547)\), and \( f(x_{\text{opt}}) = -3.86278 \)

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

with \( \alpha = (1.0, 1.2, 3.0, 3.2)^\top \) and \( A_{ij} \) and \( P_{ij} \) given in Table 1

<table>
<thead>
<tr>
<th>( A_{ij} )</th>
<th>( P_{ij} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 10 30</td>
<td>0.36890 0.1170 0.26730</td>
</tr>
<tr>
<td>0.1 10 35</td>
<td>0.46990 0.43870 0.74700</td>
</tr>
<tr>
<td>3 10 30</td>
<td>0.10910 0.87320 0.55470</td>
</tr>
<tr>
<td>0.1 10 35</td>
<td>0.03815 0.57430 0.88280</td>
</tr>
</tbody>
</table>

3. Ackley-5 function with \(-2 \leq x_i \leq 2, i = 1, \ldots, 5\), and \( (x_{\text{opt}} = 0, f(x_{\text{opt}}) = 0) \)

\[
f(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{5} \sum_{i=1}^{5} x_i^2}\right) - \exp\left(\frac{1}{5} \sum_{i=1}^{5} \cos(2\pi x_i)\right) + 20 + \exp(1).\]

References


