KATHOLIEKE UNIVERSITEIT
LEUVEN

Faculty of Business and Economics

Expected improvement in efficient global optimization through bootstrapped kriging

Inneke van Nieuwenhuyse, Jack P.C. Kleijnen and Wim van Beers

DEPARTMENT OF DECISION SCIENCES AND INFORMATION MANAGEMENT (KBI)

KBI 1022

Expected improvement in efficient global optimization through bootstrapped kriging

Inneke van Nieuwenhuyse a), Jack P.C. Kleijnen b) and Wim van Beers c)

a) Department of Decision Sciences and Information Management, K.U. Leuven, Leuven, Belgium, email : inneke.vannieuwenhuyse@econ.kuleuven.be,
b) Department of Information Management, Tilburg University, Postbox 90153, 5000 LE Tilburg, Netherlands, e-mail: kleijnen@uvt.nl,
c) Department of Quantitative Economics, University of Amsterdam, Netherlands, e-mail: W.C.M.vanBeers@uva.nl

Abstract

This paper uses a sequentialized experimental design to select simulation input combinations for global optimization, based on Kriging (also called Gaussian process or spatial correlation modeling); this Kriging is used to analyze the input/output data of the simulation model (computer code). This paper adapts the classic "expected improvement" (EI) in "efficient global optimization" (EGO) through the introduction of an unbiased estimator of the Kriging predictor variance; this estimator uses parametric bootstrapping. Classic EI and bootstrapped EI are compared through four popular test functions, including the six-hump camel-back and two Hartmann functions. These empirical results demonstrate that in some applications bootstrapped EI finds the global optimum faster than classic EI does; in general, however, the classic EI may be considered to be a robust global optimizer.

Key words: Simulation · Optimization · Kriging · Bootstrap

1 Introduction

Simulation is often used to estimate the *global optimum* of the real system being simulated (like many researchers in this area do, we use the terms "optimum" and "optimization" even if there are no constraints so the problem actually concerns minimization or maximization). The simulation model implies an input/output (I/O) function that may have multiple *local optima* (so this I/O function is not convex). Hence the major problem is that the search

may stall at such a local optimum. Solving this problem implies that the search needs to combine *exploration* and *exploitation*; i.e., the search explores the total experimental area and zooms in on the local area with the apparent global optimum—see the recent survey article [9] and the recent textbook [7] (pp. 77-107), summarized by [6].

A popular search heuristic that tries to realize this exploration and exploitation is called *EGO*, originally published by Jones, Schonlau,and Welch [12], paying tribute to earlier publications; also see [7], [8], [10], [17] (pp. 133-141), [20], [24], and the references to related approaches in [13] (pp. 154-155).

More precisely, EGO selects points (locations, input combinations) based on maximizing the *EI*. For the computation of this EI, EGO uses a *Kriging metamodel* to approximate the simulation's I/O function. Kriging metamodels are very popular in deterministic simulation, applied (for example) in engineering design; see [7] and the references in [13] (p. 3). This classic Kriging model is an exact interpolator; i.e., the Kriging predictors equal the simulated outputs observed for input combinations that have already been simulated. EGO estimates the EI through the Kriging predictor and the estimated variance of this predictor. However, Den Hertog, Kleijnen, and Siem [4] show that this classic estimator of the Kriging predictor variance is biased, and they develop an *unbiased bootstrap estimator of the Kriging predictor*. Abt [1] also points out that "considering the additional variability in the predictor due to estimating the covariance structure is of great importance and should not be neglected in practical applications". In the present article, we show that the effectiveness of EGO may be improved through the use of a bootstrapped estimator. We quantify this effectiveness through the number of simulation observations needed to reach the global optimum. Actually, our bootstrapped EI is faster in three of the four test functions; the remaining test function gives a tie. Nevertheless, the analysts may still wish to apply classic EI because they accept possible inefficiency—compared with bootstrapped EI—and prefer the simpler computations of classic EI—compared with the sampling required by bootstrapping.

Like many other authors, we assume *expensive* simulation; i.e., simulating a single point requires relatively much computer time compared with the computer time needed for fitting and analyzing a Kriging metamodel. For example, it took 36 to 160 hours of computer time for a single run of a car-crash simulation model at Ford; see [22].

We organize the remainder of this article as follows. Section 2 summarizes the *simplest* type of Kriging, but also considers the statistical complications caused by the *nonlinear* statistics in this Kriging. Section 3 summarizes *classic* EI. Section 4 adapts EI, using an unbiased *bootstrapped* estimator for the variance of the Kriging predictor. Section 5 applies the two EI variants to four popular

test functions. Section 6 presents *conclusions* and topics for *future research*. Twenty-five references conclude this article.

2 Kriging metamodels

Originally, Kriging was developed—by the South African mining engineer Daniel Krige—for interpolation in geostatistical or spatial sampling; see [3], Cressie's classic Kriging textbook. Later on, Kriging was applied to the I/O data of deterministic simulation models; see the classic article [18] and also [19], Santner et al.'s popular textbook.

Kriging may enable adequate approximation of the simulation's I/O function when the simulation experiment covers a "big" input area; i.e., the experiment is *global*, not local. "Ordinary Kriging"—simply called "Kriging " in the remainder of this article—assumes that the simulation outputs (say) w are realizations of the following *Gaussian covariance-stationary stochastic process*:

$$w(\mathbf{x}) = \mu + \delta(\mathbf{x}) \tag{1}$$

where μ denotes the constant mean and $\delta(\mathbf{x})$ has a multivariate Gaussian (Normal) distribution with mean zero and a specific covariance matrix detailed below. Kriging uses the *linear* predictor

$$y = \lambda' \mathbf{w} \tag{2}$$

where $\mathbf{w}$ denotes the vector with the n "old" simulation outputs (i.e., the outputs obtained for the n already simulated input combinations) and λ denotes the vector with the Kriging weights. To select the optimal weights in (2), Kriging uses the Best Linear Unbiased Predictor (BLUP) criterion, which minimizes the Mean Squared Error (MSE) of the predictor y. Given the assumptions of the process defined by (1), it can be proven that the *optimal Kriging weights* λ_o are

$$\lambda_o = \mathbf{\Sigma}_n^{-1}(\sigma_{n+1} + \mathbf{1}\frac{1 - \mathbf{1}'\mathbf{\Sigma}_n^{-1}\sigma_{n+1}}{\mathbf{1}'\mathbf{\Sigma}_n^{-1'}\mathbf{1}}) \tag{3}$$

where $\mathbf{\Sigma}_n = (cov(w_i, w_{i'}))$ with $i, i' = 1, \ldots, n$ denotes the $n \times n$ matrix with the covariances between the n old outputs; $\sigma_{n+1} = (cov(w_i, w_{n+1}))$ denotes the n-dimensional vector with the covariances between the n old outputs w_i and the "new" output w_{n+1} which is to be predicted; $\mathbf{1}$ denotes the n -dimensional vector with ones. Obviously, σ_{n+1} varies with the input combination of this new output, so the optimal weights λ_o are not constants.

Kriging assumes that output values $w(\mathbf{x}_g)$ and $w(\mathbf{x}_{g'})$ $(g, g' = 1, \ldots, n+1)$ are more correlated as their input locations $\mathbf{x}_g$ and $\mathbf{x}_{g'}$ are closer. Moreover, the

correlations between outputs in the k-dimensional input space are assumed to be the product of the k individual correlation functions; e.g.,

$$\exp(-\sum_{j=1}^{k}\theta_j h_j^{p_j}) = \prod_{j=1}^{k}\exp(-\theta_j h_j^{p_j}) \tag{4}$$

where h_j denotes the Euclidean distance in the j^{th} dimension of the input combinations $\mathbf{x}_g$ and $\mathbf{x}_{g'}$; θ_j denotes the importance of input dimension j (the higher θ_j is, the faster the correlation function decreases with the distance), and p_j determines the smoothness of the correlation function; e.g., $p_j = 1$ yields the exponential correlation function, and $p_j = 2$ gives the so-called Gaussian correlation function. The correlation function (4) implies that the optimal weights (3) decrease with the *distance* between the new input combination to be predicted ($\mathbf{x}_{n+1}$) and the n old combinations ($\mathbf{x}_i$, $i = 1, \ldots, n$).

EGO uses the MSE of the BLUP, which can be derived to be

$$\sigma^2(\mathbf{x}) = \sigma^2(1 - {\sigma_{n+1}}'\mathbf{\Sigma}_n^{-1}\sigma_{n+1} + \frac{(1 - \mathbf{1}'\mathbf{\Sigma}_n^{-1}\sigma_{n+1})^2}{\mathbf{1}'\mathbf{\Sigma}_n^{-1}\mathbf{1}}) \tag{5}$$

where $\sigma^2(\mathbf{x})$ denotes the variance of $y(\mathbf{x})$ (the Kriging predictor at location $\mathbf{x}$) and σ^2 denotes the (constant) variance of w, for which the covariance-stationary process (1) is assumed; a recent reference is [7] (p. 84). Note that the MSE equals the variance because the Kriging predictor is unbiased. We call $\sigma^2(\mathbf{x})$ defined in (5) the *predictor variance*.

A major problem in Kriging is that the correlation function is unknown, so both the type and the parameter values must be estimated. To estimate these parameters, the standard Kriging literature and software uses Maximum Likelihood Estimators (MLEs). The MLEs of the correlation parameters θ_j in (4) require constrained maximization, which is a hard problem because matrix inversion is necessary, the likelihood function may have multiple local maxima, etc.; see [15]. To estimate correlation functions like (4), the corresponding optimal Kriging weights (3), the resulting BLUP Kriging predictor (2), and the predictor variance (5), we use the *DACE software*, which is a free-of-charge Matlab toolbox well documented by [14]. (Alternative free software is mentioned in [13] (p. 146).)

The classic Kriging literature, software, and practice replace the optimal weights λ in (2) by the estimated optimal weights $\widehat{\lambda_0}$ which result from replacing the unknown covariances $\mathbf{\Sigma}_{i;i'}$ and $\sigma_{i;n+1}$ in (3) by their estimators $\widehat{\mathbf{\Sigma}_{i;i'}}$ and $\widehat{\sigma_{i;n+1}}$ that result from the MLEs (say) $\widehat{\sigma^2}$ and $\widehat{\theta}_j$. Unfortunately, this replacement makes the estimated optimal Kriging predictor

$$\widehat{y} = \widehat{\lambda_o}'\mathbf{w} \tag{6}$$

a *nonlinear* estimator. The classic literature ignores this complication, and simply plugs the estimates $\widehat{\sigma^2}$ and $\widehat{\theta}_j$ into the right-hand side of (5) to obtain the *estimated predictor variance*

$$s^2(\mathbf{x}) = \widehat{\sigma^2}(1 - \widehat{\sigma_{n+1}}'\widehat{\mathbf{\Sigma}_n}^{-1}\widehat{\sigma_{n+1}} + \frac{(1 - \mathbf{1}'\widehat{\mathbf{\Sigma}_n}^{-1}\widehat{\sigma_{n+1}})^2}{\mathbf{1}'\widehat{\mathbf{\Sigma}_n}^{-1}\mathbf{1}}). \tag{7}$$

It is well known that $s^2(\mathbf{x})$ is zero at the n old input locations; $s^2(\mathbf{x})$ tends to increase as the new location lies farther away from old locations. However, Den Hertog et al. [4] show that not only does $s^2(\mathbf{x})$ underestimate the true predictor variance, but the classic estimator and their unbiased bootstrapped estimator (to be detailed in Section 4) do not reach their maxima for the same input combination!

Note that in general, bootstrapping is a simple method for quantifying the behavior of nonlinear statistics; see [5], Efron and Tibshirani's classic textbook on bootstrapping. An alternative method is used in [16], to examine the consequences of estimating σ^2 and θ_j (through MLE); i.e., that article uses a first-order expansion of the MSE; earlier, [1] also used first-order Taylor series expansion. Our bootstrapped estimator is simpler and unbiased.

3 Classic EI

Forrester et al. [7] (pp. 91-106) provides a recent and in-depth discussion of classic EI (including a number of EI variations). Classic EI assumes deterministic simulation aimed at finding the unconstrained *global* minimum of the objective function, using the Kriging predictor $\widehat{y}$ and its *classic* estimated predictor variance $s^2(\mathbf{x})$ defined in (7). This EI uses the following steps, where we use our own notation distinguishing between the simulation output w and the Kriging metamodel output y.

(1) Find among the n *old* simulated outputs w_i $(i = 1, \ldots, n)$ the *minimum*, $\min_i w_i$ $(i = 1, \ldots, n)$.

(2) Estimate the input combination $\mathbf{x}$ that maximizes $\widehat{EI(\mathbf{x})}$, the estimated expected improvement over the minimum found in Step 1:

$$\widehat{EI(\mathbf{x})} = \int_{-\infty}^{\min_i w_i} [\min_i w_i - \widehat{y(\mathbf{x})}] f[\widehat{y(\mathbf{x})}] d\widehat{y(\mathbf{x})} \tag{8}$$

where $f[\widehat{y(\mathbf{x})}]$ denotes the distribution of the Kriging predictor for the input combination $\mathbf{x}$. EI assumes that this distribution is a normal distribution with the estimated mean $\widehat{y(\mathbf{x})}$ given by (6) and a variance equal

to the estimated predictor variance $s^2(\mathbf{x})$ defined in (7). To find the *maximizer* of (8), we may use either a space-filling design with candidate points or a global optimizer such as the Genetic Algorithm (GA) in [7] (p. 78).

(3) *Simulate* the maximizing combination found in Step 3 (which gives $\max_{\mathbf{x}} \widehat{EI(\mathbf{x})}$), *refit* the Kriging model to the old and new I/O data, and *return* to Step 1—unless we conclude that we have reached the global minimum *close enough* because $\max_{\mathbf{x}} \widehat{EI(\mathbf{x})}$ is "close" to zero.

Note that a *local* optimizer in Step 2 is not attractive, because $\widehat{EI(\mathbf{x})}$ is a "bumpy" function with many local optima: at all old input combinations we have $s^2(\mathbf{x}) = 0$ so $\widehat{EI(\mathbf{x})} = 0$.

4 Bootstrapped EI

Because $s^2(\mathbf{x})$ defined in (7) is an unbiased estimator of the predictor variance, we may use the *unbiased* bootstrapped estimator that was developed in [4]. That article uses *parametric bootstrapping* assuming the deterministic simulation output w forms the *Gaussian* process (1). That bootstrapping uses the MLEs of the Kriging parameters that are computed from the "original" old I/O data (say) $(\mathbf{x}_1, \mathbf{w}_1)$ with $\mathbf{x}_1$ the $n \times k$ input matrix and $\mathbf{w}_1 = (w_1, ..., w_n)'$ the corresponding output vector. We denote these MLEs by $\widehat{\mu}$, $\widehat{\sigma^2}$, and $\widehat{\theta}_j$ $(j = 1, \ldots, k)$; see the text above (6). We compute these MLE estimates through DACE (different software may give different estimates because of the difficult constrained maximization required by MLE).

Actually, [4]. gives several bootstrap algorithms. However, its first algorithm called "a fixed test set"—namely, the candidate set—gives ill conditioned matrixes in DACE. Therefore we use its second algorithm, called "adding new points one at a time": though we have many candidate points, we add a single point at a time to the old points; see Step 2 in the preceding section. Unfortunately, it turns out that this second algorithm gives bumpy plots for the bootstrapped Kriging variance as a function of a one-dimensional input (see Figure 3 in Den Hertog et al. 2006). This *bumpiness* might make our EGO approach less efficient!

Using this algorithm to estimate the MSE of the Kriging predictor at the new point $\mathbf{x}_{n+1}$, we sample (or bootstrap) *both* n old I/O data $(\mathbf{x}_1, \mathbf{w}_1^*)$ with $\mathbf{w}_1^* = (w_1^*, ..., w_n^*)'$ *and* a new point $(\mathbf{x}_{n+1}, w_{n+1}^*)$ where all $n + 1$ outputs collected in $\mathbf{w}^{*\prime} = (\mathbf{w}_1^{*\prime}, w_{n+1}^*)$ are correlated:

$$\mathbf{w}^* \sim N_{n+1}(\widehat{\mu}, \widehat{\mathbf{\Sigma}}) \tag{9}$$

with the mean vector $\widehat{\mu}$ that has all its $(n+1)$ elements equal to $\widehat{\mu}$ and the (symmetric positive-definite) $(n+1) \times (n+1)$ covariance matrix

$$\widehat{\mathbf{\Sigma}} = \begin{bmatrix} \widehat{\mathbf{\Sigma}_n} & \widehat{\sigma_{n+1}} \\ \widehat{\sigma'_{n+1}} & \widehat{\sigma^2} \end{bmatrix}$$

where (analogously to the symbols defined below (3)) $\widehat{\mathbf{\Sigma}_n}$ is the $n \times n$ matrix with the MLE of the covariances between the old outputs, $\widehat{\sigma_{n+1}}$ is the n-dimensional (column) vector with the estimated covariances between the n old outputs and the one new output, and $\widehat{\sigma^2}$ is the MLE of the constant variance σ^2 of the process specified in (1).

The bootstrapped Kriging predictor for the new point $\widehat{y^*_{n+1}}$ depends on the old I/O bootstrapped data $(\mathbf{x}_1, \mathbf{w}^*_1)$, which are used to compute the MLEs $\widehat{\mu^*}$, $\widehat{\sigma^*}$, and $\widehat{\theta^*_j}$. Note that we start our search for these $\widehat{\theta^*_j}$ with $\widehat{\theta_j}$ (the MLEs based on the original data $(\mathbf{x}_1, \mathbf{w}_1)$). The Squared Errors (SEs) at these old points are zero, because Kriging is an exact interpolator. However, the squared error at the new point is

$$SE_{n+1} = (\widehat{y^*_{n+1}} - w^*_{n+1})^2. \tag{10}$$

To reduce sampling error, we repeat this bootstrapping B times (e.g., $B = 100$), which gives $\widehat{y^*_{n+1;b}}$ with $b = 1, \ldots, B$. Combined with (10), this bootstrap sample gives the bootstrap estimator of the Kriging predictor's MSE at the new point $\mathbf{x}_{n+1}$:

$$s^2(\widehat{y^*_{n+1}}) = \frac{\sum_{b=1}^{B} (\widehat{y^*_{n+1;b}} - w^*_{n+1;b})^2}{B}. \tag{11}$$

We use this $s^2(\widehat{y^*_{n+1}})$ to compute the EI in (8) where we replace the general distribution $f[\widehat{y(\mathbf{x})}]$ by

$$N(\widehat{y_{n+1}}, s^2(\widehat{y^*_{n+1}})). \tag{12}$$

We perform the same procedure for each candidate point $\mathbf{x}_{n+1}$. To speed-up the computations of the bootstrap estimator $s^2(\widehat{y^*_{n+1}})$ in (11) for the many candidate points, we use the property that the multivariate normal distribution (9) implies that its *conditional* output is also normal. So, we still let $\mathbf{w}^*_1$ denote the bootstrapped outputs of the old input combinations; we let w^*_2 denote the output of a candidate combination. Then (9) implies that the distribution of the bootstrapped new output w^*_2—given the n bootstrapped old points $\mathbf{w}^*_1$—is (also see equation 19 in [4])

$$N(\widehat{\mu} + \widehat{\sigma'_{n+1}} \widehat{\mathbf{\Sigma}_n}^{-1} (\mathbf{w}^*_1 - \widehat{\mu}), \widehat{\sigma^2} - \widehat{\sigma'_{n+1}} \widehat{\mathbf{\Sigma}_n}^{-1} \widehat{\sigma_{n+1}}). \tag{13}$$

We interpret this formula as follows. If (say) all n elements of $\mathbf{w}^*_1 - \widehat{\mu}$ (in the first term, which represents the mean) happen to be positive, then we expect w^*_2 also to be "relatively" high ($\widehat{\sigma_{n+1}}$ has positive elements only); i.e., higher

than its unconditional mean $\widehat{\mu}$. The second term (including the variances) implies that w_2^* has a lower variance than its unconditional variance $\widehat{\sigma^2}$ if $\mathbf{w}_1$ and w_2 show high positive correlations (see $\widehat{\sigma_{n+1}}$). (The variance of w_2^* is lower than the variance of its predictor $\widehat{y_2^*}$; see [12] (equation 9).

We note that the bootstrapped predictions for all candidate points use the same bootstrapped MLEs $\widehat{\mu^*}$, $\widehat{\sigma^*}$, and $\widehat{\theta_j^*}$ computed from the n old I/O data $(\mathbf{x}_1, \mathbf{w}_1^*)$. Furthermore, to specify the initial design we use Matlab's random lhs design (which implies that each macroreplicate uses slightly different design points; we shall discuss macroreplicates in the next section). To specify the candidate points we (like many other authors) use a small space-filling design; more specifically, we use the deterministic maximin Latin hypercube designs from the website http://www.spacefillingdesigns.nl/.

5 Empirical results for four test functions

In this section, we compare the effectiveness of classic and bootstrapped EI, for *four test functions* with multiple optima; namely, Forrester et al.'s one-dimensional test function given in [7] (see Section 5.1), the two-dimensional six-hump camel-back function (Section 5.2), the three-dimensional Hartmann-3 function (Section 5.3), and the six-dimensional Hartmann-6 function (Section 5.4).

For each function, we start with an *initial design* with n_{init} points to fit an initial Kriging model. Next, we update this design sequentially, applying either classic EI or bootstrapped EI. We estimate the maximum EI through a set of n_{test} *candidate* points; the candidate point that maximizes the estimated EI is added next to the design (see step 3 in Section 3).

Because bootstrapped EI implies sampling, we repeat the experiment ten times for each test function to reduce the randomness in our results; these ten *macroreplicates* are identical except for the pseudorandom number (PRN) seed used to draw the bootstrap samples. Obviously, for classic EI a single macroreplicate suffices.

We *stop* our search when either the maximum EI is "small"—namely, $EI < e^{-20}$—or a maximum allowable number of points have been added to the initial design. For both approaches, we report the estimated optimum location x_{opt} with its objective value w_{opt}, the total number of points simulated before the heuristic stops (n_{tot}), and the iteration number that gives the estimated optimum n_{opt} (obviously, $n_{opt} \leq n_{tot}$; if the very last point simulated gives the estimated optimum, then $n_{opt} = n_{tot}$).

5.1 Forrester et al.'s test function

In [7] (pp. 83-92) Forrester et al. illustrate classic EI through the following one-dimensional test function:

$$w(x) = (6x - 2)^2 \sin(12x - 4) \text{ with } 0 \leq x \leq 1. \tag{14}$$

It can be proven that in the continuous domain, this function has one local minimum (at $x = 0.01$) and one global minimum at the input $x^o = 0.7572$ with output $w(x^o) = -6.02074$.

We use the same *initial* design as [7] does; namely, the $n_{init} = 3$ equi-spaced (or gridded) input locations 0, 0.5, and 1. The set of *candidate* points consists of a grid with distance 0.01 between consecutive input locations; this yields $n_{test} = 98$ candidate points. Given this (discrete) grid, it can be proven that the global optimum occurs at $x^o = 0.76$ with $w(x^o) = -6.0167$. The genetic algorithm in [7] finds the optimum in the continuous domain within 8 iterations, so we also set the maximum number of allowable iterations at 8. Table 1 shows the results of both EI approaches for this test function. Both approaches turn out to find the true optimum. Bootstrapped EI, however, finds this optimum faster (i.e., it requires fewer iterations) in six of the ten macroreplicates; two macroreplicates yield a tie; in the remaining two macroreplicates classic EI is faster.

Note that our results confirm the results in [4]; i.e., the classic and the bootstrapped variance of the Kriging predictor—defined in (7) and (11)—do not reach their maxima at the same input point; moreover, this classic estimator underestimates the true variance (given $n = 3$ old points). To save space, we do not display the corresponding figures.

5.2 Six-hump camel-back function

The six-hump camel-back function is defined by

$$w(x_1, x_2) = 4x_1^2 - 2.1x_1^4 + x_1^6/3 + x_1x_2 - 4x_2^2 + 4x_2^4 \tag{15}$$

with $-2 \leq x_1 \leq 2$ and $-1 \leq x_2 \leq 1$. In the continuous domain, this function has two global minima; namely, $\mathbf{x}_1^o = (0.089842, -0.712656)'$ and $\mathbf{x}_2^o = (-0.089842, 0.712656)'$ with $w(\mathbf{x}_1^o) = w(\mathbf{x}_2^o) = -1.031628$. It also has two additional local minima. For further details we refer to [23] (pp. 183-184).

We select an *initial* spacefilling design with 21 points, like Schonlau did in [21]; moreover, this selection approximates the popular rule-of-thumb that recom-

Table 1
Forrester et al.'s test function: results for bootstrapped and classic EI

Bootstrap EI macrorep.	x_{opt}	w_{opt}	n_{opt}	n_{tot}
1	0.76	-6.017	9	11
2	0.76	-6.017	10	11
3	0.76	-6.017	9	10
4	0.76	-6.017	10	10
5	0.76	-6.017	8	10
6	0.76	-6.017	11	11
7	0.76	-6.017	11	11
8	0.76	-6.017	9	10
9	0.76	-6.017	6	10
10	0.76	-6.017	9	11
Classic EI	0.76	-6.017	10	11

mends to start with a design containing $10k$ points; see [12]. More specifically, we use the maximin Latin Hypercube design found on http://www.spacefillingdesigns.nl/.

We select 200 *candidate* point through the maximin Latin hypercube design found on the same website. In this discrete set, the global minima occur at $\mathbf{x}_1^o = (-0.0302, 0.7688)'$ and $\mathbf{x}_2^o = (0.0302, -0.7688)$ with $w^o = -0.9863$. We set the maximum number of allowable iterations at 40.

Table 2 shows the results of both EI approaches for this test function. Both approaches succeed in finding the true optimum within the candidate set of points. However, bootstrapped EI finds that optimum a bit quicker, in all macroreplicates; see the column n_{opt}.

5.3 Hartman-3 function

The Hartman-3 function is given by

$$w(x_1, x_2, x_3) = -\sum_{i=1}^{4} \alpha_i \exp[-\sum_{j=1}^{3} A_{ij}(x_j - P_{ij})^2] \tag{16}$$

with parameters $\alpha = (1.0, 1.2, 3.0, 3.2)'$, and A_{ij} and P_{ij} given in Table 3; $0 \leq x_i \leq 1$ for $i = 1, 2, 3$.

Table 2
Six-hump camel-back test-function: results for bootstrapped and classic EI

Bootstrap EI macrorep.	x_{opt}	w_{opt}	n_{opt}	n_{tot}
1	(0.0302,-0.7688)	-0.9863	29	43
2	(-0.0302,0.7688)	-0.9863	29	41
3	(-0.0302,0.7688)	-0.9863	29	42
4	(0.0302,-0.7688)	-0.9863	29	42
5	(0.0302,-0.7688)	-0.9863	29	43
6	(-0.0302,0.7688)	-0.9863	25	43
7	(0.0302,-0.7688)	-0.9863	27	41
8	(0.0302,-0.7688)	-0.9863	26	42
9	(-0.0302,0.7688)	-0.9863	30	41
10	(-0.0302,0.7688)	-0.9863	26	43
Classic EI	(-0.0302,0.7688)	-0.9863	31	41

Table 3
Parameters A_{ij} and P_{ij} of the Hartman-3 function

A_{ij}	3	10	30
	0.1	10	35
	3	10	30
	0.1	10	35
P_{ij}	0.36890	0.1170	0.26730
	0.46990	0.43870	0.74700
	0.10910	0.87320	0.55470
	0.03815	0.57430	0.88280

In the continuous domain, the function has a global minimum at $\mathbf{x}^o = (0.114614, 0.555649, 0.852547)'$ with $w(\mathbf{x}^o) = -3.86278$; the function has three additional local minima.

We select an *initial* maximin Latin hypercube design with 30 points found on http://www.spacefillingdesigns.nl/, and a set of *candidate* points consisting of a maximin Latin hypercube design with 300 points generated by Matlab. In this discrete domain, the global minimum is $\mathbf{x}^o = (0.2088, 0.5465, 0.8767)'$ with $w(\mathbf{x}^o) = -3.7956$. We set the maximum allowable number of iterations

at 35.

Table 4 shows that the bootstrapped EI finds the optimum faster, in nine of the ten macroreplicates; macroreplicate 5 gives a tie.

Table 4
Hartman-3 function: results for bootstrapped and classic EI

Bootstrapped EI macrorep	x_{opt}	w_{opt}	n_{opt}	n_{tot}
1	(0.2088,0.5465,0.8767)	-3.7956	34	65
2	(0.2088,0.5465,0.8767)	-3.7956	34	65
3	(0.2088,0.5465,0.8767)	-3.7956	41	65
4	(0.2088,0.5465,0.8767)	-3.7956	34	65
5	(0.2088,0.5465,0.8767)	-3.7956	44	65
6	(0.2088,0.5465,0.8767)	-3.7956	43	65
7	(0.2088,0.5465,0.8767)	-3.7956	34	65
8	(0.2088,0.5465,0.8767)	-3.7956	34	65
9	(0.2088,0.5465,0.8767)	-3.7956	41	65
10	(0.2088,0.5465,0.8767)	-3.7956	34	65
Classic EI	(0.2088,0.5465,0.8767)	-3.7956	44	65

5.4 Hartman-6 function

The Hartman-6 function is

$$w(x_1,\ldots,x_6) = -\sum_{i=1}^{4} c_i \exp[-\sum_{j=1}^{6} \alpha_{ij}(x_j - p_{ij})^2] \tag{17}$$

with parameters $\mathbf{c} = (1.0, 1.2, 3.0, 3.2)'$, and α_{ij} and p_{ij} given in Table 5; $0 \le x_i \le 1$ $(i = 1,\ldots,6)$.

In the continuous domain, this function has a global minimum at $\mathbf{x}^o = (0.20169, 0.150011, 0.476874, 0.275332, 0.311652, 0.6573)'$ with $w(\mathbf{x}^o)$=−3.32237; the function also has five additional local minima.

We select an *initial* maximin Latin hypercube design with 51 points, like Schonlau did in [21]. Our set of *candidate* points consists of Matlab's maximin Latin hypercube design with 500 points. Within this discrete domain, the global minimum occurs at $\mathbf{x}^o = (0.3535, 0.8232, 0.8324, 0.4282, 0.1270, 0.0013)'$

Table 5
Parametrs α_{ij} and p_{ij} of the Hartman-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

with $w(\mathbf{x}^o) = -2.3643$. We set the maximum allowable number of iterations to 50.

Table 6 shows that our bootstrapped EI is faster in five of the ten macroreplicates. An explanation may be that the initial design has 51 points, which gives very many points to estimate the $k = 6$ individual correlation functions in (4) so the bias of the classic variance estimator vanishes. (An initial design size of roughly $10k$ seems necessary, because otherwise the Kriging metamodel would be too bad an approximation—even if its correlation function is estimated accurately.)

Table 6
Hartman-6 test function: results for bootstrapped and classic EI

Bootstrap EI macrorep.	x_{opt}	w_{opt}	n_{opt}	n_{tot}
1	(0.3535,0.8232,0.8324,0.4282,0.127,0.0013)	-2.3643	92	101
2	(0.3535,0.8232,0.8324,0.4282,0.127,0.0013)	-2.3643	89	101
3	(0.3535,0.8232,0.8324,0.4282,0.127,0.0013)	-2.3643	78	101
4	(0.3535,0.8232,0.8324,0.4282,0.127,0.0013)	-2.3643	86	101
5	(0.3535,0.8232,0.8324,0.4282,0.127,0.0013)	-2.3643	92	101
6	(0.3535,0.8232,0.8324,0.4282,0.127,0.0013)	-2.3643	98	101
7	(0.3535,0.8232,0.8324,0.4282,0.127,0.0013)	-2.3643	76	101
8	(0.3535,0.8232,0.8324,0.4282,0.127,0.0013)	-2.3643	78	101
9	(0.3535,0.8232,0.8324,0.4282,0.127,0.0013)	-2.3643	73	101
10	(0.3535,0.8232,0.8324,0.4282,0.127,0.0013)	-2.3643	75	101
Classic EI	(0.3535,0.8232,0.8324,0.4282,0.127,0.0013)	-2.3643	79	101

6 Conclusions and future research

In this article, we study the EI criterion in the EGO approach to global optimization. We compare the classic Kriging predictor variance estimator and our bootstrapped estimator introduced by Den Hertog et al. in [4]. For the empirical comparison of these two estimators we use four test functions, and found the following results:

(1) Forrester et al.'s one-dimensional function: Our bootstrapped EI finds the global optimum faster in six of the ten macroreplicates; two macroreplicates yield a tie; in the remaining two macroreplicates, classic EI is faster.
(2) The two-dimensional six-humped camel-back function: Our bootstrapped EI finds the global optimum quicker, in all ten macroreplicates.
(3) The Hartmann-3 function: Our bootstrap EI finds the optimum faster in nine of the ten macroreplicates; the one remaining macroreplicate gives a tie.
(4) The Hartmann-6 function: Our bootstrapped EI is faster in five of the ten macroreplicates.

Altogether, our bootstrapped EI is faster in three of the four test functions; in the remaining test function a tie occurs. Nevertheless, the analysts may apply classic EI if they accept some possible inefficiency—compared with bootstrapped EI—and prefer the simpler computations of classic EI—compared with the sampling required by bootstrapping. So we might conclude that the classic EI is a quite robust criterion.

We propose the following topics for *future research*:

- *Global convergence* of EGO; see [7] (p. 134).
- *Constrained* optimization; see [7] (pp. 125-131).
- *Random* simulation: [7] (pp. 141-153) discuss noisy simulation; i.e., numerical inaccuracy, not noise caused by pseudorandom numbers (which are used in discrete-event simulation). For the latter noise we refer to [2], [11], and [25].

Acknowledgment

We thank Emmanuel Vazquez (SUPÉLEC) for bringing Abt (1999) and Müller and Pronzato (2009) to our attention.

References

[1] Abt, M. (1999). Estimating the prediction mean squared error in Gaussian stochastic processes with exponential correlation structure. *Scandinavian Journal of Statistics*, 26, no 4, pp. 563-578

[2] Ankenman, B., B. Nelson, and J. Staum (2010), Stochastic kriging for simulation metamodeling, *Operations Research* (forthcoming)

[3] Cressie, N.A.C. (1993), *Statistics for spatial data: revised edition.* Wiley, New York

[4] Den Hertog, D., J.P.C. Kleijnen, and A.Y.D. Siem. (2006). The correct Kriging variance estimated by bootstrapping. *Journal Operational Research Society*, 57, pp. 400–409

[5] Efron, B., R.J. Tibshirani. 1993. *An introduction to the bootstrap.* Chapman & Hall, New York.

[6] Forrester, A.I.J. and A.J. Keane (2009), Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences,* 45, issue 1-3, pp. 50-79

[7] Forrester, A., A. Sóbester, and A. Keane (2008), *Engineering design via surrogate modelling: a practical guide.* Wiley, Chichester, United Kingdom

[8] Frazier, P., W. Powell, and S. Dayanik (2009), The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing* (forthcoming)

[9] Fu, M.C. 2007. Are we there yet? The marriage between simulation & optimization. *OR/MS Today*, 34, pp. 16–17

[10] Gorissen, D. (2010), *Grid-enabled adaptive surrogate modeling for computer aided engineering.* Ph.D. dissertation, Ghent University, Ghent, Belgium

[11] Huang, D., T.T. Allen, W. Notz, N. Zheng. (2006), Global optimization of stochastic black-box systems via sequential Kriging meta-models. *Journal of Global Optimization,* 34, pp. 441–466

[12] Jones, D.R., M. Schonlau, and W.J. Welch (1998), Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13, pp. 455-492

[13] Kleijnen, J.P.C. (2008), *Design and analysis of simulation experiments.* Springer

[14] Lophaven, S.N., H.B. Nielsen, and J. Sondergaard (2002). DACE: a Matlab Kriging toolbox, version 2.0. IMM Technical University of Denmark, Lyngby

[15] Martin, J.D., and T.W. Simpson (2005), On the use of Kriging models to approximate deterministic computer models. *AIAA Journal*, 43, no. 4, pp. 853-863

[16] Müller, W. G. and L. Pronzato (2009). Towards an optimal design equivalence theorem for random fields? IFAS Research Paper Series No. 2009-45, Department of Applied Statistics, Johannes Kepler University Linz, Linz, Austria

[17] Nakayama, H., Y. Yun, and M. Yoon (2009), *Sequential approximate multiobjective optimization using computational intelligence.* Springer, Berlin

[18] Sacks, J., W.J. Welch, T.J. Mitchell, and H.P. Wynn (1989), Design and analysis of computer experiments (includes Comments and Rejoinder). *Statistical Science*, 4, no. 4, pp. 409-435

[19] Santner, T.J., B.J. Williams, and W.I. Notz (2003), *The design and analysis of computer experiments.* Springer-Verlag, New York

[20] Sasena, M.J, P. Papalambros, and P. Goovaerts (2002), Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering Optimization,* 34, no.3, pp. 263-278

[21] Schonlau, M. (1997), Computer experiments and global optimization, Ph.D. thesis, University of Waterloo, Waterloo, Canada.

[22] Simpson, T.W., A.J. Booker, D. Ghosh, A.A. Giunta, P.N. Koch, and R.-J. Yang (2004), Approximation methods in multidisciplinary analysis and optimization: a panel discussion. *Structural and Multidisciplinary Optimization*, 27, no. 5, pp. 302–313

[23] Törn, A., and A. Žilinkas (1989), *Global Optimization*, Springer Verlag, Berlin

[24] Villemonteix, J., E. Vazquez, M. Sidorkiewicz, and E. Walter (2009), Global optimization of expensive-to-evaluate functions: an empirical comparison of two sampling criteria. *Journal of Global Optimization*, 43, no. 2-3, pp. 373 - 389

[25] Yin, J., S.H. Ng, and K.M. Ng (2009), A study on the effects of parameter estimation on Kriging model's prediction error in stochastic simulations. *Proceedings of the 2009 Winter Simulation Conference*, edited by M.D. Rossini, R.R. Hill, B. Johansson, A. Dunkin, and R.G. Ingalls, pp. 674-685