Expected improvement in efficient global optimization through bootstrapped kriging

**Jack P. C. Kleijnen · Wim van Beers ·
Inneke van Nieuwenhuyse**

Abstract   This article 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 design and analysis adapt the classic "expected improvement" (EI) in "efficient global optimization" (EGO) through the introduction of an improved estimator of the Kriging predictor variance; this estimator uses parametric bootstrapping. Classic EI and bootstrapped EI are compared through various test functions, including the six-hump camel-back and several 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.

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

J. P. C. Kleijnen (✉)
Department of Information Management, Tilburg University, Postbox 90153, 5000 LE Tilburg,
The Netherlands
e-mail: kleijnen@tilburguniversity.edu
URL: http://center.uvt.nl/staff/kleijnen/

W. van Beers
Department of Quantitative Economics, University of Amsterdam, Amsterdam, The Netherlands
e-mail: W.C.M.vanBeers@uva.nl

I. van Nieuwenhuyse
Department of Decision Sciences and Information Management, K.U. Leuven, Leuven, Belgium
e-mail: inneke.vannieuwenhuyse@econ.kuleuven.be

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 [12] and the recent textbook [9, pp. 77–107], summarized in [8].

A popular search heuristic that tries to realize this exploration and exploitation is called *EGO*, originally published by Jones et al.[15], paying tribute to earlier publications; also see [9,11,13,22,27,31, pp. 133–141], and the references to related approaches in [16, pp. 154–155].

More specifically, 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 [9] and the references in [16] (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 [6] show that this classic estimator of the Kriging predictor variance is biased, and they develop an *improved bootstrap estimator of the Kriging predictor variance*. 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". Moreover, the classic and the bootstrapped predictor variance do not reach their maximum at the same point (see [6]). In the present article, we demonstrate that the effectiveness of EGO may indeed be improved through the use of this bootstrapped estimator. We quantify this effectiveness through the number of simulation observations needed to reach the global optimum. We find that this 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.

We conjecture that our bootstrapped estimator of the variance of the biased (nonlinear) Kriging predictor defined in (4) (which uses Maximum Likelihood Estimators or MLEs of the Kriging parameters) is *unbiased*, but we have no mathematical proof. We base our conjecture on the general bootstrap theory in [7]. A simple example is the biased estimator s of the standard deviation σ of a Gaussian (Normal) variable (s^2 is an unbiased estimator of σ^2); we conjecture that bootstrapping may give an unbiased estimator of the variance of this biased estimator s.

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 36–160 h of computer time for a single run of a car-crash simulation model at Ford; see [29]. In such an expensive simulation, the initial sample size might be selected to be "small", given the number of dimensions (number of inputs to be optimized) and the shape of the I/O function implied by the specific simulation model. Unfortunately, a too small sample may give a Kriging metamodel that is inadequate to guide the search for the global optimum—using EGO or some other heuristic. We shall briefly return to this problem later on in this article.

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 predictor. Section 3 summarizes *classic* EI. Section 4 adapts EI, using a *bootstrapped* estimator for the variance of the Kriging predictor. Section 5 gives *numerical* results, first for the classic and the bootstrapped *variance* estimators in a simple test function; next for the two *EI* variants in four popular test functions. Section 6 presents *conclusions* and topics for *future research*. Thirty-two 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 the classic Kriging textbook [4]. Later on, Kriging was applied to the I/O data of deterministic simulation models; see the classic article [25] and also the popular textbook [26].

Kriging may enable adequate approximation of the simulation's I/O function, even 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 function being studied is a realization of a *Gaussian stochastic process* $Y(\mathbf{x}) = \mu + Z(\mathbf{x})$ where $\mathbf{x}$ is a point in a d-dimensional search space, μ is its constant mean, and $Z(\mathbf{x})$ is a zero-mean, stationary, Gaussian stochastic process with variance σ^2 and some assumed correlation function such as

$$corr[Y(\mathbf{x}_i), Y(\mathbf{x}_j)] = \prod_{k=1}^{d} \exp\left(-\theta_k \left|x_{ik} - x_{jk}\right|^{p_k}\right), \tag{1}$$

which implies that the correlations between outputs in the d-dimensional input space are the product of the d individual correlation functions. Such a correlation function implies that outputs $Y(\mathbf{x}_i)$ and $Y(\mathbf{x}_j)$ are more correlated as their input locations $\mathbf{x}_i$ and $\mathbf{x}_j$ are "closer"; i.e., their Euclidean distance in the kth dimension of the input combinations $\mathbf{x}_i$ and $\mathbf{x}_j$ is smaller. The correlation parameter θ_k denotes the importance of input dimension k (the higher θ_k is, the faster the correlation function decreases with the distance), and p_k determines the smoothness of the correlation function; e.g., $p_k = 1$ yields the exponential correlation function, and $p_k = 2$ gives the so-called Gaussian correlation function. Realizations of such a Gaussian process are smooth, continuous functions; its specific behavior in terms of smoothness and variability along the coordinate directions is determined by the parameters μ, σ^2, and $\theta = (\theta_1, \ldots \theta_d)'$.

Given a set of (say) n "old" observations $\mathbf{y} = (y_1, \ldots, y_n)'$, Kriging uses the Best Linear Unbiased Predictor (BLUP) criterion—which minimizes the Mean Squared Error (MSE) of the predictor—to derive the following *linear* predictor for a point $\mathbf{x}_{n+1}$, which may be either a new or an old point:

$$\widehat{y}(\mathbf{x}_{n+1}) = \widehat{\mu} + \mathbf{r}' \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\widehat{\mu}) \tag{2}$$

where $\widehat{\mu} = \mathbf{1}'\mathbf{R}\mathbf{y}/\mathbf{1}'\mathbf{R}\mathbf{1}$ and $\mathbf{1}$ denotes the n-dimensional vector with ones, $\mathbf{R}$ the $n \times n$ correlation matrix whose (i,j)th entry is given by (1), and $\mathbf{r} = \{corr[Y(\mathbf{x}), Y(\mathbf{x}_1)], \ldots, corr[Y(\mathbf{x}), Y(\mathbf{x}_n)]\}'$ the vector of correlations between $\mathbf{x}$ and the n sampled points. It can be proven that if $\mathbf{x}_{n+1}$ is an old point, then the predictor equals the observed value $(\widehat{y}(\mathbf{x}) = y(\mathbf{x}))$; i.e., the Kriging predictor is an *exact interpolator*.

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

$$\sigma^2(\mathbf{x}) = \sigma^2 \left(1 - \mathbf{r}'\mathbf{R}^{-1}\mathbf{r} + \frac{(1 - \mathbf{1}'\mathbf{R}^{-1}\mathbf{r})^2}{\mathbf{1}'\mathbf{R}^{-1}\mathbf{1}} \right) \qquad (3)$$

where $\sigma^2(\mathbf{x})$ denotes the variance of $\widehat{y}(\mathbf{x})$ (the Kriging predictor at location $\mathbf{x}$) and σ^2 denotes the (constant) variance of Y, for which a covariance-stationary process is assumed; a recent reference is [9] (p. 84). We call $\sigma^2(\mathbf{x})$ defined in (3) 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. Like most simulation studies (unlike geostatistical studies) assume, we assume a Gaussian correlation function (so $p_k = 2$ in Eq. 1). To estimate the parameters of this correlation function, the standard Kriging literature and software uses MLEs. The MLEs of the correlation parameters θ_k in (1) require constrained maximization, which is a hard problem because matrix inversion is necessary, the likelihood function may have multiple local maxima, etc.; see [20]. To estimate the resulting Kriging predictor (2), and the predictor variance (3), we use the *DACE software*, which is a free-of-charge Matlab toolbox well documented in [19]. (Alternative free software is mentioned in [10] and [16, p. 146].)

The classic Kriging literature, software, and practice replace the unknown covariances $\mathbf{R}$ and $\mathbf{r}$ in (2) and (3) by their estimators $\widehat{\mathbf{R}}$ and $\widehat{\mathbf{r}}$ that result from the MLEs (say) $\widehat{\boldsymbol{\psi}} = (\widehat{\mu}, \widehat{\sigma^2}, \widehat{\boldsymbol{\theta}}')'$. Unfortunately, this replacement changes the linear predictor $\widehat{y}(\mathbf{x})$ defined in (2) into the *nonlinear* predictor

$$\widehat{\widehat{y}}(\mathbf{x}_{n+1}) = \widehat{\mu} + \widehat{\mathbf{r}}'\widehat{\mathbf{R}}^{-1}(\mathbf{y} - \mathbf{1}\widehat{\mu}). \qquad (4)$$

The classic literature ignores this complication, and simply plugs the estimates $\widehat{\sigma^2}$ and $\widehat{\boldsymbol{\theta}}_j$ into the right-hand side of (3) to obtain the *estimated predictor variance* of $\widehat{\widehat{y}}(\mathbf{x})$:

$$s^2(\mathbf{x}) = \widehat{\sigma^2}(1 - \widehat{\mathbf{r}}'\widehat{\mathbf{R}}^{-1}\widehat{\mathbf{r}} + \frac{(1 - \mathbf{1}'\widehat{\mathbf{R}}^{-1}\widehat{\mathbf{r}})^2}{\mathbf{1}'\widehat{\mathbf{R}}^{-1}\mathbf{1}}). \qquad (5)$$

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. [6] show that not only does $s^2(\mathbf{x})$ *underestimate* the true predictor variance, but the classic estimator and their bootstrapped estimator (to be detailed in Sect. 4) do not reach their *maxima* at the same input combination!

In general, *bootstrapping* is a simple method for quantifying the behavior of nonlinear statistics; see the classic textbook on bootstrapping [7]. An alternative method is used in [21], to examine the consequences of estimating σ^2 and $\widehat{\boldsymbol{\theta}}$ (through MLE); i.e., that article uses a first-order expansion of the MSE; earlier, Abt [1] also used first-order Taylor series expansion. One more alternative method developed in [32] uses the Bayesian approach. Our specific bootstrapped estimator is simpler.

3 Classic EI

A recent and in-depth discussion of classic EI is [9, pp. 91–106] (also discussing 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 $\widehat{\widehat{y}}$ and its *classic* estimated predictor variance $s^2(\mathbf{x})$ defined in (4) and (5). This EI uses the following steps.

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

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

$$\widehat{EI(\mathbf{x})} = \int_{-\infty}^{\min_i y_i} \left[\min_i y_i - y(\mathbf{x}) \right] f\left[y(\mathbf{x}) \right] dy(\mathbf{x}) \tag{6}$$

where $f[y(\mathbf{x})]$ denotes the distribution of $\widehat{\widehat{y}}(\mathbf{x})$ (the Kriging predictor with MLEs for the input combination $\mathbf{x}$). EI assumes that this distribution is a Gaussian distribution with the estimated mean $\widehat{\widehat{y}}(\mathbf{x})$ and a variance equal to the estimated predictor variance $s^2(\mathbf{x})$. To find the *maximizer* of (6), we may use either a space-filling design with *candidate* points or a *global optimizer* such as the genetic algorithm in [9, p. 78].

(3) Simulate the maximizing combination found in Step 2 (which gave $\max_{\mathbf{x}} \widehat{EI(\mathbf{x})}$), *refit* the Kriging model to the old and new I/O data, and *return* to Step 1—unless the conclusion is that the global minimum is reached close enough because $\max_{\mathbf{x}} \widehat{EI(\mathbf{x})}$ is "close" to zero.

Note that a *local* optimizer in Step 2 is undesirable, because $\widehat{EI(\mathbf{x})}$ has many local optima; i.e., for all old input combinations $s^2(\mathbf{x}) = 0$ so $\widehat{EI(\mathbf{x})} = 0$.

4 Bootstrapped EI

Because $s^2(\mathbf{x})$ defined in (5) is a biased estimator of the predictor variance, we may use the bootstrapped estimator that was developed in [6]. That article uses *parametric bootstrapping* assuming the deterministic simulation outputs Y are realizations of a *Gaussian* process, as explained in Sect. 2. That bootstrapping computes $\widehat{\psi}$, the MLEs of the Kriging parameters $\left(\widehat{\mu}, \widehat{\sigma^2}, \widehat{\theta}' \right)'$, from the "original" old I/O data $(\mathbf{x}, \mathbf{y})$ defined in Section 2 (so $\mathbf{x}$ is the $n \times d$ input matrix and $\mathbf{y} = (y_1, \ldots, y_n)'$ is the corresponding output vector). We compute these MLEs through DACE (different software may give different estimates because of the difficult constrained maximization required by MLE). These MLEs specify the distribution from which we will sample so-called *bootstrapped* observations; actually, this so-called *parametric* bootstrapping is no more than Monte Carlo sampling from a given type of distribution with parameter values estimated from the original data. (There is also nonparametric bootstrapping, which is relevant in random simulation with replicates; [17] applies such bootstrapping for constrained optimization in random simulation.)

Note that [6] gives several bootstrap algorithms; we use its second algorithm. Unfortunately, [6] finds that this 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 second 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 $(\mathbf{x}, \mathbf{y}^*)$ with $\mathbf{y}^* = (y_1^*, \ldots, y_n^*)'$ *and* a new point $(\mathbf{x}_{n+1}, y_{n+1}^*)$ where all $n + 1$ outputs collected in $\mathbf{y}_{n+1}^{*'} = (\mathbf{y}^{*'}, y_{n+1}^*)$ are correlated:

$$\mathbf{y}_{n+1}^* \sim N_{n+1}(\widehat{\mu}_{n+1}, \widehat{\Sigma}_{n+1}) \tag{7}$$

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

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

The bootstrapped Kriging predictor for the new point $\left(\widehat{\widehat{y}}^*_{n+1} \right)$ depends on the bootstrapped function values $\mathbf{y}^*$ at the old $\mathbf{x}$ data, which are used to compute the bootstrapped MLEs $\widehat{\boldsymbol{\psi}}^* = \left(\widehat{\mu}^*, \widehat{\sigma}^{2*}, \widehat{\boldsymbol{\theta}}^{*\prime} \right)'$. Note that we start our search for these $\widehat{\boldsymbol{\theta}}^*$ with $\widehat{\boldsymbol{\theta}}$ (the MLEs based on the original data $(\mathbf{x}, \mathbf{y})$).

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

$$SE_{n+1} = (\widehat{\widehat{y}}^*_{n+1} - y^*_{n+1})^2. \tag{8}$$

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

$$s^2(\widehat{\widehat{y}}^*_{n+1}) = \frac{\sum\limits_{b=1}^{B} (\widehat{\widehat{y}}^*_{n+1,b} - y^*_{n+1;b})^2}{B}. \tag{9}$$

We use this $s^2(\widehat{\widehat{y}}^*_{n+1})$ to compute the EI in (6) where we replace the general distribution $f[\widehat{y}(\mathbf{x})]$ by

$$N\left[\widehat{\widehat{y}}_{n+1}, s^2(\widehat{\widehat{y}}^*_{n+1}) \right]. \tag{10}$$

We perform the same procedure for each candidate point $\mathbf{x}_{n+1}$.

To avoid repeated factorization of $\widehat{\Sigma}_{n+1}$ in (7) for the many candidate points, we use the property that the multivariate normal distribution (7) implies that its *conditional* output is also normal. So, we still let $\mathbf{y}^*$ denote the bootstrapped outputs at the n old input combinations, and y^*_{n+1} denote the bootstrapped output of a candidate combination. Then (7) implies that the distribution of this y^*_{n+1} at a specific candidate point $\mathbf{x}_{n+1}$—given (or "conditional on") the n bootstrapped outputs $\mathbf{y}^*$—is (also see [14, p. 157] and equation 19 in [6])

$$N\left(\widehat{\mu} + \widehat{\mathbf{r}}' \widehat{\mathbf{R}}^{-1} \left(\mathbf{y}^* - \widehat{\mu} \right), \widehat{\sigma^2} \left(1 - \widehat{\mathbf{r}}' \widehat{\mathbf{R}}^{-1} \widehat{\mathbf{r}} \right) \right). \tag{11}$$

We interpret this formula as follows. If (say) all n elements of $\mathbf{y}^* - \widehat{\mu}$ (in the first term, which represents the mean) happen to be positive, then we expect y^*_{n+1} also to be "relatively" high ($\widehat{\mathbf{r}}$ has positive elements only); i.e., higher than its unconditional mean $\widehat{\mu}$. The second term (including the variances) implies that y^*_{n+1} has a lower variance than its unconditional variance $\widehat{\sigma^2}$ if $\mathbf{y}$ and y_{n+1} show high positive correlations (see $\widehat{\mathbf{r}}$). (The variance of y^*_{n+1} is lower than the variance of its predictor $\widehat{\widehat{y}}^*_{n+1}$; see [15], Eq. 9.)

We note that the bootstrapped predictions for all candidate points use the same bootstrapped MLEs $\widehat{\boldsymbol{\psi}}^*$ computed from the n bootstrapped old I/O data $(\mathbf{x}, \mathbf{y}^*)$.

5 Empirical results

In this section, we first compare the *coverage* (or "success rate") of 90% confidence intervals (CIs) for the classic predictor variance versus our bootstrapped predictor variance, in an example that guarantees that the Kriging metamodel is an adequate approximation (see Sect. 5.1). Next 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 [9], the two-dimensional six-hump camel-back function, the three-dimensional Hartmann-3 function, and the six-dimensional Hartmann-6 function (see Sect. 5.2).

5.1 Coverage rates in a Kriging model

In this subsection we estimate the *coverage rates* of 90% CIs; i.e., do these intervals indeed have a 90% probability of covering the true value? Note that the classic method and our method use the same point predictor but different estimated variances; see (4), (5), and (10). We construct an example guaranteeing that the Kriging metamodel is an adequate (valid) approximation, as follows.

We decide to select $d = 2$ dimensions, and generate (sample) observations from a Gaussian process with parameters $\mu = 3.3749$, $\sigma^2 = 0.0176$, $\theta_1 = 0.1562$, and $\theta_1 = 2.5$ (we select these values after fitting a Kriging model to the camel-back function defined in Sect. 5.2.2 with $n = 2,601$ I/O observations). Now we generate (say) T sample paths over a 51×51 grid with $-0.5 \leq x_1 \leq 0.5$ and $0 \leq x_2 \leq 1$ (yielding 2,601 equally spaced points). From this grid we sample n points to act as the initial sample and another point to act as the point $\mathbf{x}_{n+1}$ to be predicted. For each sample path $t = 1, \ldots, T$ we apply both the classic predictor variance and our bootstrapped variance to make a 90% CI for the prediction at $\mathbf{x}_{n+1}$. If the CI covers the actual value from the sampled path, we call it a "success". The goal of this experiment is to estimate the effects of n (initial sample size) on the coverage, so we select n equal to 5, 20, 50, and 80 respectively. Table 1 shows the results of the classic and the bootstrap approaches with mean coverage rates estimated from 2,601-n test points. Though some test points yield higher coverage rates for the classic approach (these individual results are not displayed), this table suggests

- in both approaches, the mean coverage rates increase as the initial sample increases;
- our bootstrap gives higher mean coverage rates for any sample size;
- the difference between the mean coverage rates decreases as the sample size increases;
- our mean coverage rate is close to the nominal prespecified value if the sample size agrees with the rule-of-thumb, $n = 10d$ (so $n = 20$), proposed in [15] and [18].

In the appendix we also estimate the coverages for the function $y = \sin(x)$; we find similar results.

So our conclusion is that our bootstrapped variance estimator of the Kriging predictor tends to exceed the classic estimator, so the mean coverage rates are higher for our bootstrapped

Table 1 Gaussian Process test function: mean coverage rates of 90% confidence intervals with classic and bootstrapped predictor variance

n	5	20	50	80
Classic	0.7198	0.8065	0.8637	0.8866
Bootstrap	0.7643	0.8459	0.8747	0.8903

estimator; nevertheless, for "small" initial samples these higher coverage rates are still much lower than the prespecified values, because the initial Kriging metamodel is an inadequate approximation of the true I/O function.

5.2 Effectiveness in four test functions

In this section we compare classic and bootstrapped EI through four test functions; namely, the function in [9], the so-called six-hump camel function, the Hartmann-3 function, and the Hartmann-6 function.

For each of these four functions, we start with an *initial design* with n 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 Sect. 3).

Because bootstrapped EI implies sampling, we repeat the experiment ten times for each test function to reduce the randomness in our results; by definition, 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 y_{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 $n_{opt} = n_{tot}$).

5.2.1 Forrester et al.'s function

In [9, pp. 83–92] classic EI is illustrated through the following one-dimensional function:

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

It can be proven that in the continuous domain, this function has one local minimum (at $x = 0.01$) and one global minimum at $x^o = 0.7572$ with output $y(x^o) = -6.02074$. We use the same *initial* design as [9] does; namely, the $n = 3$ 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 $y(x^o) = -6.0167$. The genetic algorithm in [9] finds the optimum in the continuous domain after 8 iterations, so we also set the maximum number of allowable iterations at 8. Table 2 shows the results of both EI approaches for this 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 [6]; i.e., the classic and the bootstrapped variance of the Kriging predictor—defined in (5) and (9)—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.

Table 2 Forrester et al.'s function: classic versus bootstrapped EI

	x_{opt}	y_{opt}	n_{opt}	n_{tot}
Classic EI	0.76	−6.017	10	11
Bootstrap EI				
Macroreplicate				
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

5.2.2 Six-hump camel-back function

The six-hump camel-back function is defined by

$$y(x_1, x_2) = 4x_1^2 - 2.1x_1^4 + x_1^6/3 + x_1 x_2 - 4x_2^2 + 4x_2^4 \tag{13}$$

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 $y(\mathbf{x}_1^o) = y(\mathbf{x}_2^o) = -1.031628$. It also has two additional local minima. For further details we refer to [30, pp. 183–184].

We select an *initial* spacefilling design with $n = 21$ points, like Schonlau did in [28]; moreover, this selection approximates the popular rule-of-thumb $n = 10d$. More specifically, we use the maximin Latin Hypercube Sampling (LHS) design found on http://www.spacefillingdesigns.nl/, which compares various designs and collects the best designs on this website.

We select 200 *candidate* points through the maximin LHS 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 $y^o = -0.9863$. We set the maximum number of allowable iterations at 40.

Table 3 shows the results of both EI approaches. Both approaches succeed in finding the true optimum within the candidate set of points. However, our bootstrapped EI finds that optimum a bit quicker, in all macroreplicates; see the column n_{opt}.

5.2.3 Hartmann-3 function

The Hartmann-3 function is defined by

$$y(x_1, x_2, x_3) = -\sum_{i=1}^{4} \alpha_i \exp\left[-\sum_{j=1}^{3} A_{ij}(x_j - P_{ij})^2 \right] \tag{14}$$

Table 3 Six-hump camel-back function: classic versus bootstrapped EI

	x_{opt}	y_{opt}	n_{opt}	n_{tot}
Classic EI	$(-0.0302, 0.7688)$	-0.9863	31	41
Bootstrap EI				
Macrorep.				
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

Table 4 Parameters A_{ij} and P_{ij} of the Hartmann-3 function

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

with $0 \leq x_i \leq 1$ for $i = 1, 2, 3$; parameters $\alpha = (1.0, 1.2, 3.0, 3.2)'$, and A_{ij} and P_{ij} given in Table 4. In the continuous domain, the function has a global minimum at $\mathbf{x}^o = (0.114614, 0.555649, 0.852547)'$ with $y(\mathbf{x}^o) = -3.86278$; the function has three additional local minima.

We select an *initial* maximin LHS design with 30 points found on http://www.spacefillingdesigns.nl/, and a set of *candidate* points consisting of a maximin LHS 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 $y(\mathbf{x}^o) = -3.7956$. We set the maximum allowable number of iterations at 35. Table 5 shows that the bootstrapped EI finds the optimum faster, in nine of the ten macroreplicates; macroreplicate 5 gives a tie.

5.2.4 Hartmann-6 function

The Hartmann-6 function is defined by

$$y(x_1, \ldots, x_6) = -\sum_{i=1}^{4} c_i \exp\left[-\sum_{j=1}^{6} \alpha_{ij}(x_j - p_{ij})^2 \right] \tag{15}$$

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

Table 5 Hartmann-3 function: classic versus bootstrapped EI

	x_{opt}	y_{opt}	n_{opt}	n_{tot}
Classic EI	(0.2088, 0.5465, 0.8767)	−3.7956	44	65
Bootstrapped EI				
Macrorep.				
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

Table 6 Parameters α_{ij} and p_{ij} of the 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

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 $y(\mathbf{x}^o) = -3.32237$; the function also has five additional local minima.

We select an *initial* maximin LHS design with 51 points, as in [28]. Our set of *candidate* points consists of Matlab's maximin LHS design with 500 points. Within this discrete domain, the global minimum occurs at $\mathbf{x}^o = (0.3535, 0.8232, 0.8324, 0.4282, 0.1270, 0.0013)'$ with $y(\mathbf{x}^o) = -2.3643$. For the maximum allowable number of iterations we select 50.

Table 7 shows that our bootstrapped EI is faster in only five of the ten macroreplicates. An explanation may be that the initial design has 51 points; the noncollapsing property of LHS means that projection onto any axis gives an approximately equally spread sample of points on that axis. Hence, accurate estimation of the correlation function in that dimension is possible for the $k = 6$ individual correlation functions in (1) so the bias of the classic variance estimator vanishes. (An initial design size of roughly $10d$ seems necessary, because otherwise the Kriging metamodel may be too bad an approximation—even if its correlation function is estimated accurately. For the camel-back and Hartmann-3 functions we also use

Table 7 Hartmann-6 function: classic versus bootstrapped EI

	x_{opt}	y_{opt}	n_{opt}	n_{tot}
Classic EI	(0.3535,0.8232,0.8324,0.4282,0.127,0.0013)	−2.3643	79	101
Bootstrap EI				
Macrorep.				
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

approximately $n = 10d$, but d is then only 2 and 3 respectively so the individual correlation functions are estimated from smaller samples.)

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 [6]. We estimate the effects of the *initial* sample size on the difference between the classic and the bootstrapped estimates of the predictor variance. These empirical results suggest that the smaller that sample size is, the more the classic estimator underestimates the true predictor variance. Unfortunately, a "small" sample size—given the number of dimensions and the (unknown) shape of the I/O function—increases the likelihood of an inadequate Kriging metamodel so the Kriging (point) predictor may be misleading; i.e., this wrong predictor combined with a correct predictor variance may give a wrong EI leading to the (expensive) simulation of the wrong next point.

To compare EI combined with the classic and the bootstrapped variance estimators empirically, we use four test functions, and find 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) Six-hump camel-back function in two-dimensions: Our bootstrapped EI finds the global optimum quicker, in all ten macroreplicates.

(3) Hartmann-3 function: Our bootstrap EI finds the optimum faster in nine of the ten macroreplicates; the one remaining macroreplicate gives a tie.

(4) Hartmann-6 function: Our bootstrapped EI is faster in five of the ten macroreplicates.

Altogether, our bootstrapped EI is better in three of the four test functions; the remaining test function gives a tie. Nevertheless, the analysts might wish to apply classic EI because

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 gives a quite *robust* heuristic. One explanation of this robustness may be that the bias of the classic variance estimator decreases as the sample size increases so this estimator approaches the bootstrapped estimator (both approaches use the same point predictor).

We propose the following topics for *future research*:

- Testing the adequacy (validity) of the Kriging metamodel; see [17].
- *Global convergence* of EGO; see [3, p. 134] and [9].
- *Constrained* optimization; see [9, pp. 125–131].
- *Random* simulation: [9, pp. 141–153] discusses numerical noise, not noise caused by pseudorandom numbers (which are used in discrete-event simulation). For the latter noise we refer to [2], [23], and [32].
- Application to *large-scale* industrial problems, such as the so-called MOPTA08 problem with 124 inputs and 68 inequality constraints for the outputs; see [24]
- Comparison of EGO with *other* approaches; see [5].

Appendix: Test function $y = \sin(x)$

Consider the following example: $y = \sin(x)$ with $0 \leq x \leq 30$. First consider the *uniform* design with $n = 5$ points that all happen to give $y = \max[\sin(x)] = 1$ so $y_1 = \cdots = y_5 = 1$. Hence Kriging gives $\widehat{\mu} = 1$ and $\widehat{\sigma^2} = 0$, so $\widehat{\widehat{y}} = 1$ and $s^2(\mathbf{x}) = 0$. Obviously these Kriging results are very misleading!

Next consider a design with the following $n = 7$ randomly selected x-values: 0.25, 3.00, 6.00, 10.15, 16. 80, 23.48, and 29.00. This design also gives a bad Kriging metamodel, which underestimates the predictor variance. These two examples demonstrate that some form of validation of the metamodel seems useful; see the "future research topics" in Sect. 6.

However, the bad results of these two examples might also be mitigated through the application of the general *randomization principle* that is advocated in design of experiments (DOE); i.e., randomization may avoid pathological phenomena. In our case we propose *random LHS* design; i.e., we sample n values for x using Matlab's *maximin lhs* macro. To select the new point x_{n+1}, we take a grid with step size 0.05 so we get 601 points. Because

Table 8 $y = \sin(x)$ test function: mean coverage rates of 90% confidence intervals with classic and bootstrapped predictor variance

n	4	6	8	10	20	80
Classic	0.5451	0.6438	0.9291	0.9368	1.000	0.8866
Bootstrap	0.5825	0.6648	0.9470	0.9443	0.9999	0.8903

random LHS implies sampling error, we repeat the whole experiment 30 times (so the number of "macroreplicates" is 30). This gives Table 8. We find the same pattern as we do for the Gaussian process in Sect. 5.1:

- in both approaches, the mean overage rates increase as the initial sample increases;
- our bootstrap gives higher mean coverage rates, until the sample becomes "very large";
- the difference between the mean coverage rates decreases, as the sample size increases;
- if the initial sample size agrees with the rule-of-thumb $n = 10d$, then the mean coverage rate is acceptable.

References

1. Abt, M.: Estimating the prediction mean squared error in Gaussian stochastic processes with exponential correlation structure. Scand. J. Stat. **26**(4), 563–578 (1999)
2. Ankenman, B., Nelson, B., Staum, J.: Stochastic kriging for simulation metamodeling. Oper. Res. **58**(2), 371–382 (2010)
3. Bull, A.D.: Convergence Rates of Efficient Global Optimization algorithms. Arxiv (2011)
4. Cressie, N.A.C.: Statistics for Spatial Data: Revised Edition. Wiley, New York (1993)
5. del Castillo, E., Santiago, E.: A Matrix-T approach to the sequential design of optimization experiments. IIE Trans. (accepted 2010)
6. Den Hertog, D., Kleijnen Siem, A.Y.D.: The correct kriging variance estimated by bootstrapping. J. Oper. Res. Soc. **57**, 400–409 (2006)
7. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman & Hall, New York (1993)
8. Forrester, A.I.J., Keane, A.J.: Recent advances in surrogate-based optimization. Prog. Aearosp. Sci. **45**(1–3), 50–79 (2009)
9. Forrester, A., Sóbester, A., Keane, A.: Engineering Design via Surrogate Modelling: a Practical Guide. Wiley, Chichester (2008)
10. Frazier, P.I. : Learning with Dynamic Programming. In: Cochran, J.J., Cox, L.A., Keskinocak, P., Kharoufeh, J.P., Smith, J.C. (eds.) Wiley Encyclopedia of Operations Research and Management Science, Wiley, New York (2011)
11. Frazier, P., Powell, W., Dayanik, S.: The knowledge-gradient policy for correlated normal beliefs. INFORMS J. Comput. (2009)
12. Fu, M.C.: Are we there yet? The marriage between simulation & optimization. OR/MS Today **34**, 16–17 (2007)
13. Gorissen, D.: Grid-enabled Adaptive Surrogate Modeling for Computer Aided Engineering. Ph.D. dissertation, Ghent University, Ghent, Belgium (2010)
14. Hardle, W., Simar, L.: Applied Multivariate Statistical Analysis. Springer, New York (2003)
15. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. J. Global Optim. **13**, 455–492 (1998)
16. Kleijnen, J.P.C.: Design and Analysis of Simulation Experiments.Springer (Chinese translation: published by Publishing House of Electronics Industry, Beijing, 2010) (2008)
17. Kleijnen, J.P.C., Van Beers, W., Van Nieuwenhuyse, I.: Constrained optimization in simulation: a novel approach. EJOR **202**, 164–174 (2010)
18. Loeppky J., L., Sacks, J., Welch, W.: Choosing the sample size of a computer experiment: a practical guide. Technometrics **51**, 366–376 (2009)
19. Lophaven, S.N., Nielsen, H.B., Sondergaard, J.: DACE: a Matlab Kriging Toolbox, Version 2.0. IMM Technical University of Denmark, Lyngby (2002)
20. Martin, J.D., Simpson, T.W.: On the use of kriging models to approximate deterministic computer models. AIAA J. **43**(4), 853–863 (2005)
21. Müller, W. G., Pronzato, L.: 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 (2009)
22. Nakayama, H., Yun, Y., Yoon, M.: Sequential Approximate Multiobjective Optimization Using Computational Intelligence. Springer, Berlin (2009)
23. Picheny, V., Ginsbourger, D., Richet, Y.: Noisy Expected Improvement and on-line computation time allocation for the optimization of simulators with tunable fidelity. In: 2nd International Conference on Engineering Optimization, September 6–9, 2010, Lisbon, Portugal (2010)

24. Regis, R.G.: Stochastic radial basis function algorithms for large-scale optimization involving expensive black-box objective and constraint functions. Comput. & OR **38**, 837–853 (2011)

25. Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design and analysis of computer experiments (includes comments and rejoinder). Stat. Sci. **4**(4), 409–435 (1989)

26. Santner, T.J., Williams, B.J., Notz, W.I.: The Design and Analysis of Computer Experiments. Springer, New York (2003)

27. Sasena, M.J, Papalambros, P., Goovaerts, P.: Exploration of metamodeling sampling criteria for constrained global optimization. Eng. Optimiz. **34**(3), 263–278 (2002)

28. Schonlau, M.: Computer Experiments and Global Optimization, Ph.D. thesis, University of Waterloo, Waterloo, Canada (1997)

29. Simpson, T.W., Booker, A.J., Ghosh, D., Giunta, A.A., Koch, P.N., Yang, R.-J.: Approximation methods in multidisciplinary analysis and optimization: a panel discussion. Struct. Multidiscip. O. **27**(5), 302–313 (2004)

30. Törn, A., Žilinkas, A.: Global Optimization. Springer, Berlin (1989)

31. Villemonteix, J., Vazquez, E., Sidorkiewicz, M., Walter, E.: Global optimization of expensive-to-evaluate functions: an empirical comparison of two sampling criteria. J. Global Optim **43**(2–3), 373–389 (2009)

32. Yin, J., Ng, S.H., Ng, K.M.: A Bayesian metamodeling approach for stochastic simulations. In: Johansson, B., Jain, S., Montoya-Torres, J., Hugan, J., Yücesan, E. (eds.), Pres. 2010 WSC, pp. 1055–1066 (2010)