Simulation optimization via bootstrapped Kriging: survey

Jack P.C. Kleijnen
Department of Information Management / CentER, Tilburg University, Postbox 90153, 5000 LE Tilburg, Netherlands, Kleijnen@tilburguniversity.edu

Abstract

This contribution covers deterministic and random (stochastic) simulators, focusing on optimization via Kriging (Gaussian Process or spatial correlation) metamodels (surrogates or emulators). These Kriging models may be analyzed through bootstrapping, which is a versatile statistical method but this method must be adapted to the specific problem being analyzed. More precisely, a random simulator may be run several times for the same scenario (combination of values for the simulator's inputs); the resulting replicated responses may be resampled with replacement, which is called "distribution-free bootstrapping". A deterministic simulator, however, is run only once for the same scenario, so parametric bootstrapping is used assuming a multivariate Gaussian distribution; this distribution is sampled after its (hyper)parameters are estimated from the simulator's Input/Output data. More specifically, this contribution surveys: (1) Kriging for Efficient Global Optimization (EGO) via Expected Improvement (EI) using parametric bootstrapping to obtain an unbiased estimator of the Kriging predictor's variance accounting for the randomness resulting from estimating the Kriging parameters. (2) Distribution-free bootstrapping to validate the Kriging models in constrained optimization via Mathematical Programming. (3) Kriging metamodels with distribution-free bootstrapping for "robust" optimization which accounts for an environment that is not exactly known (so it is uncertain); this optimization uses Mathematical Programming to estimate the Pareto frontier. (4) Kriging with distribution-free bootstrapping to preserve the assumed monotonicity of the outputs (responses) as a function of the inputs.

Version: 29 April 2011

1 Introduction

In this contribution, I discuss simulation models that are either deterministic or random; a specific type of random simulators represent Discrete Event Dynamic Systems (DEDS); e.g., queueing systems. I focus on optimization of the simulated system. I assume that this optimization uses metamodels—also called surrogates or emulators—because the simulation requires much computer time; see the survey by Barton and Meckesheimer (2006). I focus on metamodels that use Kriging (Gaussian Process or GP, spatial correlation). These Kriging models may be analyzed through bootstrapping. Bootstrapping is a general versatile statistical method, for which software is available in many statistical packages including the BOOT macro in SAS and the bootstrap command in S-Plus; see Novikov and Oberman (2007); my co-authors implement bootstrapping in Matlab. Nevertheless, bootstrapping must be adapted to the specific problem being investigated. More precisely, a random simulator may be run several times for the same "scenario" ("design point" or combination of values for the simulator's inputs), which gives Identically and Independently Distributed (IID) responses. These responses may be resampled with replacement, which is called distribution-free or nonparametric bootstrapping. A deterministic simulator, however, is run only once for the same scenario, so distribution-free bootstrapping is impossible; however, parametric bootstrapping is possible because Kriging assumes a multivariate Gaussian distribution, which can be sampled—once the parameters of that distribution (or Kriging hyperparameters) are estimated from the given simulation Input/Output (I/O) data.

More specifically, I discuss the following five topics.

1. Parametric bootstrapping to obtain an unbiased estimator of the variance of the Kriging predictor when that predictor uses estimated Kriging parameters (namely, the GP's mean vector and the covariance matrix). See Section 4.

2. Use of the bootstrapped variance estimator (derived in Section 4) in Efficient Global Optimization (EGO) via Expected Improvement (EI). See Section 5.

3. Distribution-free bootstrapping in Kriging for constrained optimization via Integer Non-Linear Programming (INLP) in discrete-event simulation (e.g., a call-center simulation). See Section 6.

4. Distribution-free bootstrapping quantifying the randomness of Kriging metamodels in discrete-event simulation models for robust

optimization—in the sense of Taguchi, accounting for an environment that is not exactly known. See Section 7.

5. Monotonicity-preserving Kriging through distribution-free bootstrapping in discrete-event simulation (e.g., queueing and inventory simulations). See Section 8.

Though the idea of bootstrapping is simple, its application may require the "art" of modeling. I shall first illustrate the basic bootstrap concept through distribution-free bootstrapping in random simulation with multiple replications of scenarios; next I shall illustrate the idea through parametric bootstrapping in deterministic simulation (where replications are useless) or in random simulation with only a few replications.

Note that "random" simulation may be interpreted in three ways:

- The simulator is deterministic, but has *numerical* noise (caused by numerical approximations); see Forrester, Sóbester, and Keane (2008, p. 141).

- The simulator is deterministic, but inputs have uncertain values so these values are sampled from a prior input distribution; so-called *uncertainty propagation*. This uncertainty is called "epistemic". See Helton and Davis (2003) and Janusevskis and Le Riche (2010).

- The simulator uses *Pseudo-Random Numbers* (PRNs); e.g., queueing simulators (applied in traffic, telecommunications, supply chains) are "discrete event" simulation models that sample the occurrence of events such as the arrival and service of automobiles, telephone calls, production orders. These events may form a Poisson process, but more complicated processes are possible. This uncertainty is called 'aleatory". The most popular textbook on discrete-event simulation is Law (2007). Only a few studies consider discrete-event simulations with uncertain parameters; e.g., the Poisson parameter is uncertain (so epistemic and aleatory uncertainties are combined); see Kleijnen (2007).

I assume that the reader is familiar with the basics of Kriging, so I only summarize these basics—mainly to define symbols and terminology; see Section 2. I summarize the basics of bootstrapping in Section 3. The other sections have already been mentioned above, except for Section 9, which gives conclusions. The many references should enable the reader to further study simulation optimization via bootstrapped Kriging.

2 Kriging: basics

Originally, Kriging was developed—by the South African mining engineer *Daniel Krige*—for interpolation in geostatistical or spatial sampling; see the classic Kriging textbook Cressie (1993). Later on, Kriging was applied to the I/O data of deterministic simulation models; see the classic article Sacks et al. (1989) and also the popular textbooks Santner, Williams, and Notz (2003), and Forrester et al. (2008).

The literature on Kriging is vast, and covers diverse disciplines (e.g., statistics, mechanical engineering, operations research); e.g., the following website consists of 33 printed pages emphasizing machine learning:

http://www.gaussianprocess.org/.

My own view on Kriging is based on the publications mentioned in the first paragraph of this section; other researchers refer to the books by Rasmussen and Williams (2006) and Stein (1999).

There is much *software* implementing Kriging; see e.g. the preceding textbooks and also

http://www.gaussianprocess.org/;

my coauthors and I, however, use DACE, which is a free-of-charge Matlab-toolbox well documented in Lophaven, Nielsen, and Sondergaard (2002); alternative free software is mentioned in Frazier (2011) and Kleijnen (2008, p. 146).

Many publications focus on *expensive simulation*; i.e., simulation of a single scenario (input combination) requires relative much computer time. Some other publications focus on problems caused by large I/O data sets, so the matrix inversions in Kriging (see the equations 3 and 4 below) become problematic; also see "approximations' on

http://www.gaussianprocess.org/.

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*

$$Y(\mathbf{x}) = \mu + Z(\mathbf{x}) \tag{1}$$

where $\mathbf{x}$ is a point in a d -dimensional search space, μ is its constant mean, and $Z(\mathbf{x})$ is a stationary Gaussian stochastic process with mean zero, variance σ^2, and some assumed correlation function such as

$$corr[Y(\mathbf{x}_i),Y(\mathbf{x}_j)] = \prod_{k=1}^{d} \exp(-\theta_k \left|x_{ik} - x_{jk}\right|^{p_k}), \tag{2}$$

which implies that the correlations between outputs in this input space are the product of the d individual correlation functions. A correlation function like (2) 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 k^{th} 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 Kriging parameters μ, σ^2, and $\boldsymbol{\theta} = (\theta_1, \ldots \theta_d)'$. (Nearly all my Kriging publications use the Gaussian correlation function.)

Given a set of 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:

$$\widehat{y}(\mathbf{x}_{n+1}) = \mu + \mathbf{r}'\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\mu) \tag{3}$$

where $\mathbf{r} = \{corr[Y(\mathbf{x}_{n+1}), Y(\mathbf{x}_1)], \ldots, corr[Y(\mathbf{x}_{n+1}), Y(\mathbf{x}_n)]\}'$ is the vector of correlations between the outputs at the new point $\mathbf{x}_{n+1}$ and the n old points $\mathbf{x}_i$, $\mathbf{R}$ is the $n \times n$ matrix whose $(i, j)^{th}$ entry is given by (2), and $\mathbf{1}$ denotes the n -dimensional vector with ones; obviously the correlation vector and matrix $\mathbf{r}$ and $\mathbf{R}$ may be replaced by the corresponding covariance vector and matrix (say) $\boldsymbol{\Sigma}_{n+1} = \sigma^2\mathbf{r}$ and $\boldsymbol{\Sigma} = \sigma^2\mathbf{R}$ because σ^2 then cancels out in (3). It can be proven that if $\mathbf{x}_{n+1}$ coincides with an old point $\mathbf{x}_i$, then the predictor equals the observed value ($\widehat{y}(\mathbf{x}_i) = y(\mathbf{x}_i)$); i.e., the Kriging predictor is an *exact interpolator*.

The MSE of the BLUP can be derived to be

$$\sigma^2(\mathbf{x}) = \sigma^2(1 - \mathbf{r}'\mathbf{R}^{-1}\mathbf{r} + \frac{(1 - \mathbf{1}'\mathbf{R}^{-1}\mathbf{r})^2}{\mathbf{1}'\mathbf{R}^{-1}\mathbf{1}}) \tag{4}$$

where $\sigma^2(\mathbf{x})$ denotes the variance of $\widehat{y}(\mathbf{x})$ (the Kriging predictor at point $\mathbf{x}$; see (3)) (σ^2 still denotes the constant variance of Y, for which a covariance-stationary process is assumed); a recent reference is Forrester et al. (2008, p. 84). Because the Kriging predictor is unbiased, the MSE equals the variance. This $\sigma^2(\mathbf{x})$ may be called the *predictor variance*.

In *random* simulation the interpolation property is not desirable, so the Kriging metamodel is slightly changed including *nugget* effects or *intrinsic* noise; see Santner et al. (2003, pp. 215-249), Forrester et al. (2008, p. 143), Yin, Ng, and Ng (2009), Chen, Ankenman, and (2010). The resulting "stochastic Kriging" does not interpolate the n outputs averaged over the m_i ($i = 1, \ldots, n$) replicates for input combination i.

Chen et al. (2010) also account for correlations between the outputs for different input combinations caused by Common Random Numbers (CRN).

More precisely, the classic geostatistical literature also accounts for measurement errors, and the deterministic simulation literature also accounts for numerical noise. This extension implies that (1) is augmented with a *white noise* term (say) e:

$$Y(\mathbf{x}) = \mu + Z(\mathbf{x}) + e \tag{5}$$

where e is Normally IID (NIID) with zero mean and constant variance σ_e^2. Chen et al. (2010) generalize this white noise such that e has a variance that depends on the input combination $\mathbf{x}_i$ (variance heterogeneity) and the noise terms at various input combinations are not independent because of CRN; i.e., the covariance matrix of e (say) $\mathbf{\Sigma_e}$ does no longer equal $\sigma_e^2\mathbf{I}$ (white noise) but becomes a covariance matrix with heterogeneous variances on the main diagonal (also see Yin et al. (2009) and positive covariances off this diagonal. The Kriging predictor (3) then becomes

$$\widehat{y}(\mathbf{x}_{n+1}) = \mu + \mathbf{\Sigma}'_{n+1}(\mathbf{\Sigma} + \mathbf{\Sigma}_{\overline{e}})^{-1}(\overline{\mathbf{y}} - \mathbf{1}\mu) \tag{6}$$

where $\mathbf{\Sigma}_{\overline{e}}$ is the covariance matrix of $\overline{e} = \sum_{j=1}^{m_i} e_{i;j}/m_i$ and $\overline{\mathbf{y}}$ is the n-dimensional vector with the output averages $\overline{y_i} = \sum_{j=1}^{m_i} y_{i;j}/m_i$ computed from the m_i replicates at point i ($i = 1, , n$). A similar substitution with $\mathbf{\Sigma}_e = c\mathbf{I}$ is used to solve numerical problems in the computation of $\mathbf{R}^{-1}$; see Lophaven et al. 2003b, p. 12.).

A major problem in Kriging is that correlation functions such as (2) are *unknown*, so both the type and the parameter values must be estimated. Most simulation studies assume a Gaussian correlation function (so $p_k = 2$ in (2)); geostatistical studies, however, often assume different functions. To estimate the parameters of this correlation function, the standard Kriging literature and software uses Maximum Likelihood Estimators (MLEs). The MLEs of the correlation parameters θ_k in (2) require constrained maximization, which is a hard problem because matrix inversion is necessary, the likelihood function may have multiple local maxima, etc.; see Martin and Simpson (2005).

The classic Kriging literature, software, and practice replace the unknown $\mathbf{R}$ and $\mathbf{r}$ in (3) and (4) 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, substitution of $\widehat{\mu}$, $\widehat{\mathbf{r}}$, and $\widehat{\mathbf{R}}$ into (3) changes the linear predictor $\widehat{y}(\mathbf{x})$ into the *nonlinear* predictor (say) $\widehat{\widehat{y}}(\mathbf{x}_{n+1})$ (with double hats). The classic literature ignores this complication, and simply plugs the estimates $\widehat{\sigma^2}$, $\widehat{\mathbf{r}}$, $\widehat{\mathbf{R}}$, and $\widehat{\theta}_j$ into the right-hand side of (4) to obtain (say) $s^2(\mathbf{x})$, the *estimated*

predictor variance of $\widehat{y}(\mathbf{x})$. There is abundant Kriging software for the estimation of the resulting (classic deterministic) Kriging predictor (3), and the predictor variance (4) (see again the references in the beginning of this section).

In *random* simulation with m_i replications, the main-diagonal elements of $\widehat{\mathbf{\Sigma_e}}$ may be the unbiased estimators

$$s_i^2 = \frac{\sum_{j=1}^{m_i}(y_{i;j} - \overline{y_i})^2}{(m_i - 1)m_i} \ (i = 1, \ldots, n). \tag{7}$$

When using CRN with a constant number of replicates $m_i = m$, the off-diagonal elements of $\widehat{\mathbf{\Sigma_e}}$ are then

$$s_{i;i'} = \frac{\sum_{j=1}^{m}(y_{i;j} - \overline{y_i})(y_{i';j} - \overline{y_{i'}})}{(m-1)m} \ (i, i' = 1, \ldots, n). \tag{8}$$

From Dykstra (1970) it follows that the $n \times n$ matrix $\widehat{\mathbf{\Sigma_e}}$ is singular if $m \leq n$.

As far as software is concerned, Han and Santner (2008) provide a Matlab code for the white noise case, while Santner et al. (2003, pp. 215-249) provide a C code for this same case. Matlab code for Chen et al.'s metamodel is provided on the web; see

http://www.stochastickriging.net/.

Besides ordinary Kriging, Chen et al. also examine universal stochastic Kriging, which replaces the constant μ by a low-degree polynomial.

It is well known that in deterministic Kriging the estimated predictor variance $s^2(\mathbf{x})$ 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, Kleijnen, and Siem (2006) 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* at the same input combination! In general, *bootstrapping* is a simple method for quantifying the behavior of nonlinear statistics (such as $\widehat{y}(\mathbf{x})$); see the classic textbook on bootstrapping Efron and Tibshirani (1993) and the next section.

3 Bootstrapping: basics

Bootstrapping is a *data driven* method, so let's assume that a data set is given (say) $y_1, \ldots y_n$where the n elements y_i $(i = 1, \ldots, n)$ are IID. Consider the following two examples:

1. The y_i are not normally but *exponentially* distributed: $y_i \sim Exp(\lambda)$.

2. The statistic of interest in *nonlinear*; e.g., the estimated skewness $\sum_{i=1}^{n}(y_i - \overline{y})^3/[(n-1)s^3]$ with sample average $\overline{y}$ and sample standard deviation s.

Suppose that in Example 1 the interest is in the distribution of the sample average $\overline{y}$. If the y_i were NIID—denoted as $y_i \sim NIID(\mu, \sigma)$—then the average would have a t_{n-1} distribution. In this example, however, $y_i \sim Exp(\lambda)$; then we may estimate the parameter λ from y_i; e.g., $\widehat{\lambda} = 1/\overline{y}$. Next we can sample n new observations (say) y_i^* from $Exp(\widehat{\lambda})$: so-called *parametric bootstrapping*, which is a type of *Monte Carlo* sampling; the superscript $*$ is the usual symbol to denote bootstrapped observations. From these bootstrapped observations y_i^* we compute the statistic of interest, $\overline{y^*} = \sum_{i=1}^{n} y_i^*/n$. To estimate the Empirical density Function (EDF) of $\overline{y^*}$, we repeat this resampling (say) B times, where B is called the "bootstrap sample size"; a typical value is $B = 100$. If we wish to estimate a (say) 90% Confidence Interval (CI) for the population mean $E(y)$, then the simplest method uses the "order statistics" $y_{(i)}^*$ (so $y_{(1)}^* < y_{(2)}^* < \ldots < y_{(n)}^*$); i.e., this CI is $(y_{(0.05B)}^*, y_{(0.95B)}^*)$, assuming $0.05B$ and $0.95B$ are integers (otherwise rounding is necessary); see Efron and Tibshirani (1993, pp. 170-174).

Now suppose we do not know which type of distribution y_i has (and n is too small for a reliable estimate of the type of distribution). Then we can apply *distribution-free* or *nonparametric bootstrapping*, as follows. We resample the n "original" observations y_i with replacement (so y_1 could be sampled zero times, or only once, or even n times; if y_1 is sampled n times, then obviously none of the other values is resampled). From these resampled (bootstrapped) observations y_i^* we compute the statistic of interest, $\overline{y^*} = \sum_{i=1}^{n} y_i^*/n$. Like in parametric bootstrapping, we can compute the EDF of $\overline{y^*}$ through resampling B times;this gives the CI.

Obviously, we can apply bootstrapping to estimate the EDF of more complicated statistics than the average; e.g., the skewness in Example 2—or the Kriging predictor with an estimated covariance matrix so the predictor becomes nonlinear (see the next section). I myself have applied bootstrapping to a large variety of problems; e.g., optimization of simulated systems (see the next sections), validation of simulation models (see Kleijnen, Cheng, and Bettonvil 2001) and metamodels (see Section 6), and ranking of scientific journals on quality (see Kleijnen and Van Groenendaal 2000).

4 Bootstrapped variance of Kriging predictor

Because $s^2(\mathbf{x})$ (estimated predictor variance in deterministic Kriging, defined in Section 2) is biased, Den Hertog et al. (2006) derive an *unbiased* bootstrapped estimator. Those authors use *parametric bootstrapping* assuming the deterministic simulation outputs Y are realizations of a *Gaussian* process defined in (1). That bootstrapping first computes (say) $\widehat{\boldsymbol{\psi}} = (\widehat{\mu}, \widehat{\sigma^2}, \widehat{\theta}')'$, the MLEs of the Kriging parameters from the "original" old I/O data $(\mathbf{x}, \mathbf{y})$ (so $\mathbf{x}$ is the $n \times d$ input matrix and $\mathbf{y} = (y_1, ..., y_n)'$ is the corresponding output vector; $\widehat{\boldsymbol{\psi}}$ implies $\widehat{\mathbf{R}}$ and $\widehat{\mathbf{r}}$). Den Hertog et al. 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 to 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.

ΣNote that Den Hertog et al.'s bootstrap algorithm called "adding new points one at a time": considers many prediction points, a single point is added (one at a time) to the old n points. Unfortunately, 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).

To estimate the MSE of the Kriging predictor at the new point $\mathbf{x}_{n+1}$, sample (or bootstrap) *both* the n old I/O data $(\mathbf{x}, \mathbf{y}^*)$ with $\mathbf{y}^* = (y_1^*, ..., y_n^*)'$ *and* the new point $(\mathbf{x}_{n+1}, y_{n+1}^*)$ where all $n+1$ outputs collected in $\mathbf{y}_{n+1}^{*\prime} = (\mathbf{y}^{*\prime}, y_{n+1}^*)$ are correlated:

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

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 equals

$$\begin{bmatrix} \widehat{\boldsymbol{\Sigma}} & \widehat{\boldsymbol{\Sigma}_{n+1}} \\ \widehat{\boldsymbol{\Sigma}_{n+1}}' & \widehat{\sigma^2} \end{bmatrix}.$$

The n bootstrapped old I/O data $(\mathbf{x}, \mathbf{y}^*)$ resulting from (9) give the bootstrapped MLEs $\widehat{\boldsymbol{\psi}}^* = (\widehat{\mu}^*, \widehat{\sigma^2}^*, \widehat{\boldsymbol{\theta}}^{*\prime})'$ (Den Hertog et al. start their search for these $\widehat{\boldsymbol{\theta}}^*$ from $\widehat{\boldsymbol{\theta}}$, the MLEs based on the original data $(\mathbf{x}, \mathbf{y})$).These MLEs $\widehat{\boldsymbol{\psi}}^*$ give the bootstrapped Kriging predictor for the new point, $\widehat{\widehat{y}}_{n+1}^*$.

The Squared Errors (SEs) at these old points are zero, because classic Kriging is an exact interpolator. However, the squared error at the new

point is

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

where y^*_{n+1} results from (9).

To reduce sampling error, this bootstrapping is repeated B times (e.g., $B = 100$), which gives $\widehat{\widehat{y}}^*_{n+1,b}$ and $y^*_{n+1;b}$ with $b = 1, \ldots, B$. This gives *the* bootstrap estimator of the Kriging predictor's MSE or variance:

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

Den Hertog et al. give several examples; namely four explicit mathematical functions in one or two dimensions, and one circuit-simulator taken from Sacks et al. (1989) with $n = 32$ and $d = 6$. The bootstrap estimator may also be used in EGO, as explained next.

5 EGO with bootstrapped variance

The classic reference for EGO is Jones, Schonlau, and Welch (1998); a recent and in-depth discussion of classic EGO is Forrester et al. (2008, pp. 90-101). Classic EGO 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 Section 2. This EI uses the following steps to balance local and global search, also called *exploitation* and *exploration*.

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

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

$$\max_{\mathbf{x}} \widehat{EI(\mathbf{x})} = \int_{-\infty}^{\min_i y_i} [\min_i y_i - y(\mathbf{x})] f[y(\mathbf{x})] dy(\mathbf{x}) \tag{12}$$

   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 Gaussian with estimated mean $\widehat{\widehat{y}}(\mathbf{x})$ and variance $s^2(\mathbf{x})$. To find the *maximizer* of (12), we may use either a space-filling design with *candidate* points or a *global optimizer* such as the genetic algorithm in Forrester et al. (p. 78).

3. Simulate the maximizing combination found in Step 2, *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 $\widehat{EI(\mathbf{x})}$ is a "bumpy" function with many local optima; i.e., for all old input combinations $s^2(\mathbf{x}) = 0$ so $\widehat{EI(\mathbf{x})} = 0$.

Kleijnen, Van Beers, and Van Nieuwenhuyse (2011) use the unbiased bootstrap $s^2(\widehat{\widehat{y}}^*_{n+1})$ defined in (11) to compute the EI in (12); i.e., they replace the general distribution $f[\widehat{\widehat{y}}(\mathbf{x})]$ by $N[\widehat{\widehat{y}}_{n+1}, s^2(\widehat{\widehat{y}}^*_{n+1})]$. They perform the same procedure for each candidate point $\mathbf{x}_{n+1}$. To speed-up the computations of $s^2(\widehat{\widehat{y}}^*_{n+1})$ for the many candidate points, they use the property that the multivariate normal distribution (9) implies that its *conditional* output is also normal. So, let $\mathbf{y}^*$ still denote the bootstrapped outputs of the n old input combinations, and y^*_{n+1} the bootstrapped output of a candidate combination. Then (9) implies that the distribution of this y^*_{n+1}—given (or "conditional on") $\mathbf{y}^*$—is (also see equation 19 in Den Hertog et al.)

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

This formula may be interpreted as follows. If (say) all n elements of $\mathbf{y}^* - \widehat{\boldsymbol{\mu}}$ (see the first term, representing the mean) happen to be positive, then it may be expected that y^*_{n+1} is also "relatively" high ($\widehat{\mathbf{r}}$ has positive elements only); i.e., higher than its unconditional mean $\widehat{\mu}$. The second term 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 Jones et al. (1998, equation 9). 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 $(\mathbf{x}, \mathbf{y}^*)$.

Kleijnen et al. (2011) estimate the effects of the sample size n on the difference between the classic and the bootstrapped estimates of the predictor variance. Their empirical results suggest that the smaller n is, the more the classic estimator underestimates the true variance. Unfortunately, a "small" n—given the number of dimensions d and the (unknown) shape of the I/O function—increases the likelihood of an inadequate Kriging metamodel so the Kriging (point) predictor $\widehat{\widehat{y}}(\mathbf{x})$ 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, Kleijnen et al. (2011) use four test functions with d equal to 1,2,3, and 6. The bootstrapped EI turns out to be better in three of the four test functions; the remaining test function gives a tie. Nevertheless, the analysts might wish to stick to the 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 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 unbiased bootstrapped estimator (both approaches use the same point predictor).

Classic EGO assumes deterministic simulation aimed at finding the unconstrained global minimum of the objective function. Recently several publications relaxed this EGO assumption; see Forrester et al. (2008, pp. 125-131, 141-153), Frazier, Powell, and Dayanik (2009), Picheny, Ginsbourger, and Richet (2010), and Villemonteix, Vazquez, and Walter (2009). An approach not guided by EGO is presented in the next section.

6 Constrained optimization in random simulation

Kleijnen, Van Beers, and Van Nieuwenhuyse (2010) present a new heuristic for constrained optimization of random simulation models; i.e., they select one output (e.g., $E(y_0)$) as the objective to be minimized while the other $r-1$ outputs must satisfy prespecified threshold values c_h $(h = 1, \ldots, r-1)$ and the d simulation inputs x_j $(j = 1, \ldots, d)$ must satisfy s (linear or nonlinear) constraints f_g (e.g., budget constraints) and belong to the set of non-negative integers $\mathbf{N}$:

$$\begin{aligned} Min_{\mathbf{x}} E(y_0) \\ E(y_h) \geq c_h \ (h = 1, \ldots, r-1) \\ f_g(x_1, \ldots, x_d) \geq c_g \ (g = 1, \ldots, s) \\ x_j \in \mathbf{N} \ (j = 1, \ldots, d). \end{aligned}$$

To solve this type of problems, they develop a heuristic combining

- sequentialized Design Of Experiments (DOE) to specify the next simulation input combination (EGO has the same goal);
- Kriging to analyze the simulation I/O data (like EGO);
- Integer Non-Linear Programming (INLP) to estimate the optimal solution from the Kriging metamodels.

More specifically, they apply Kriging to the average output per simulated input combination, and do so for each of the r types of output (no multivariate Kriging). They select the number of replicates m_i such that the halfwidth of the 90% confidence interval for the average simulation output $\overline{y_h(\mathbf{d}_i)}$ is within 15% of the true mean for all r outputs. Law (2007, pp. 500-503); moreover, they apply CRN to obtain acceptable signal/noise (Chen et al. (2010) derive stochastic Kriging for random simulation with CRN). The various steps of their heuristic are summarized in Figure 1 (with $a = 30$)

The heuristic sequentially updates the initial design; i.e., it adds additional points in steps 5 and 9 respectively (a similar sequential approach is proposed in Bates et al., 2006). Points in step 5 are meant to improve the metamodel; points in step 9 are meant to find the optimum—similar to "exploration" and "exploitation" in EGO and in several other discrete-event simulation optimization heuristics surveyed in Fu (2007).

The (global) Kriging metamodel should be accurate enough to enable the INLP solver to identify clearly infeasible points (which violate the constraints on random simulation outputs) and suboptimal points (for which the goal output values are too high). Consequently, the heuristic may add points throughout the entire input-feasible area, and the metamodels use all I/O values. To guide the INLP search, the heuristic simulates each point with a specific precision to be reasonably certain of the objective values and the (non)violation of the constraints.

Step 4 applies the following six (sub)steps for *cross-validation* including *bootstrapping*; also see Kleijnen (2008):

1. From the set of simulation I/O data, delete one input combination at a time—but avoid extrapolation so (say) n_{cv} can be deleted (as detailed in Kleijnen et al. 2010).

2. Based on the remaining I/O data, compute $y_h^-(\mathbf{x}_i)$, the Kriging predictor for output h of the deleted input combination i (do not re-estimate the correlation functions defined in (2), as the current estimates based on all observations are more reliable; also see Jones et al. 1998 and Joseph, Hung, and Sudjianto 2008.)

3. Use bootstrapping to obtain $\widehat{var(y_h^*(\mathbf{x}_i))}$, the unbiased estimator for the predictor variance for output h at the deleted combination $\mathbf{x}_i$. *Distribution-free bootstrapping* might follow Van Beers and Kleijnen (2008), were it not for three complications: (i) every replicate of a given point $\mathbf{x}_i$ gives a *multi*variate output vector; (ii) *CRN* make the output vectors of the same replicate of the simulated input combinations positively correlated (also see Kleijnen

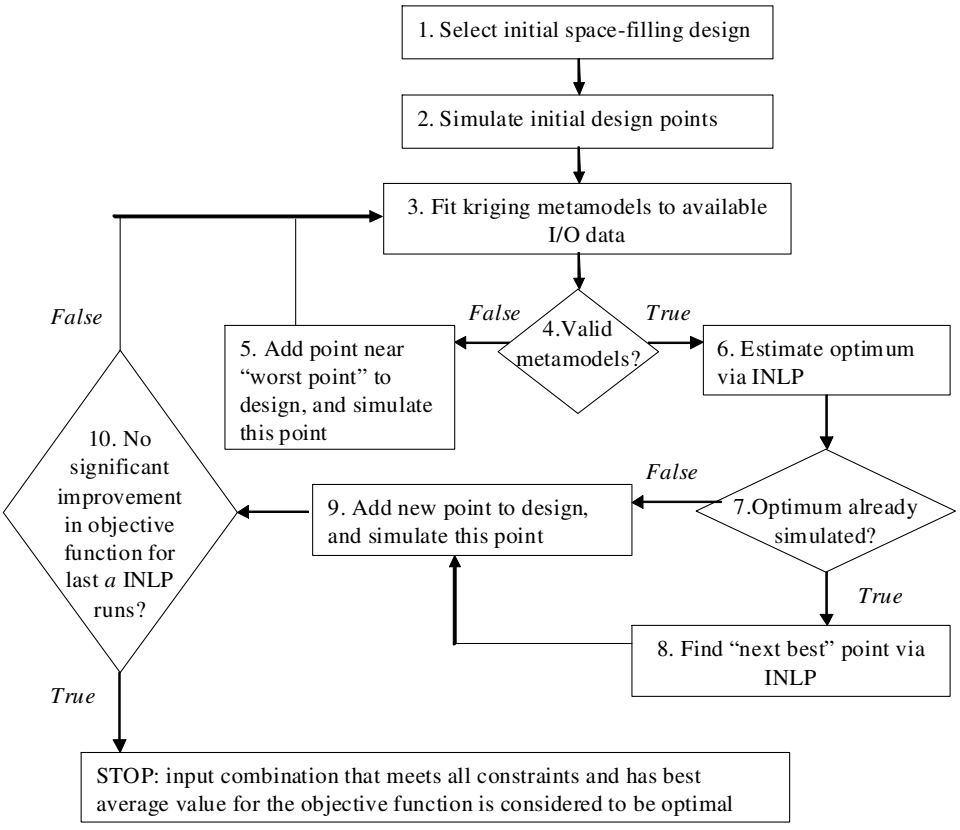

Figure 1: Overview of Kleijnen, Van Beers, and Van Nieuwenhuyse (2010)'s heuristic

and Deflandre, 2006); (iii) the number of replicates m_i may *vary* with the input combination i, when the relative precision requirement is used. See Kleijnen et al. (2010)and also Section 7.2.

4. Use these $\widehat{var(y_h^*(\mathbf{x}_i))}$ to compute the Studentized prediction errors (say) $t_{m_i-1}^{h,i}$ for every output h of every deleted combination i:

$$t_{m_i-1}^{h,i} = \frac{\overline{y_h(\mathbf{x}_i)} - y_h^-(\mathbf{x}_i)}{\sqrt{\widehat{var(y_h(\mathbf{x}_i))} + \widehat{var(y_h^*(\mathbf{x}_i))}}} \; (h = 0, \ldots, r-1) \; (i = 1, \ldots, n_{cv}) \tag{14}$$

where $\widehat{var(\overline{y_h(\mathbf{x}_i)})} = s_i^2$, which follows from (7).

5. Repeat the preceding four (sub)steps, until all n_{cv} combinations have been deleted one-at-a-time.

6. Determine the highest absolute value of the observed $t_{m_i-1}^{h,i}$ over all r outputs and all n_{cv} cross-validated input combinations, and determine if this value is statistically significant using Bonferroni's inequality, which implies that the traditional type-I error rate α is divided by $r \times n_{cv}$. If this is the case, then all r Kriging models are rejected (simulation gives estimates of all r outputs; the heuristic then moves to Step 5 in Figure 1). Else, the metamodels are considered to be valid (so they are ready for the INLP procedure in Step 6 of Figure 1).

Step 5 in Figure 1 augments the design with a new combination to improve the (global) Kriging models, if the Kriging models are rejected. The maximum Studentized prediction error is given by the so-called "worst point", so in the neighborhood of that point the heuristic requires extra information about the metamodels' behavior. However, Kriging assumes that input combinations near each other have outputs with high positive correlations, so simulating a point close to the worst point (or any other point in the current design) would give little new information. Therefore the heuristic selects the point halfway the worst point and its nearest neighbor in the current design (selecting a point "halfway" resembles EGO).

Step 6 in Figure 1 uses a free Matlab branch-and-bound INLP solver called *bnb20.m*. A disadvantage of this solver is that it guarantees only local optimality, so it needs multiple starting points; the heuristic uses three starting points (EGO is a global optimizer, but does not consider constraints on the multivariate random outputs and the multiple deterministic inputs).

In Step 6, INLP may give a previously simulated point as the optimum. In that case, Step 8 reruns the INLP solver with the additional constraint that it should return a point that is not yet part of the design; this point is called the "next best point". Mathematically, if $\mathbf{x}_t$ $(t = 1, \ldots, T)$ denotes the T points (in the d-dimensional space) that have already been simulated during the previous iterations, then impose the following T additional constraints

$$1 \leq \sum_{j=1}^{d} (x_{t;j} - x_j)^2 \ (t = 1, \ldots, T). \tag{15}$$

The next best point may be located far away from the old optimum.

The current heuristic is composed of modules that use free off-the-shelf software. These components may be replaced as the knowledge in DOE, Kriging, and INLP evolves. Kriging may be replaced by other types of metamodels; e.g., radial basis functions (see Regis 2011). Applications may have continuous inputs, so INLP may be replaced by a solver that uses the gradients, for which Kriging gives estimates "for free". The heuristic may also be adapted to deterministic simulations with constrained multiple outputs and inputs.

The heuristic is applied to Bashyam and Fu (1998)'s academic (s, S) inventory system and Kelton, Sadowski, and Sturrock (2007)'s realistic call-center simulation. The results are compared with those of the popular commercial heuristic OptQuest embedded in the Arena discrete-event simulation software (see Kelton et al. 2007); the new heuristic outperforms OptQuest in terms of number of simulated input combinations and quality of the estimated optimum.

7 Robust optimization in simulation

In practice, at least some inputs of a given simulation model are uncertain so the optimum solution that is derived ignoring these uncertainties may be wrong. Decision-making in such an uncertain world may use Taguchi (1987)'s approach, originally developed to help Toyota design "robust" cars; i.e., cars that perform reasonably well in many circumstances (ranging from polar to desert environments).

Dellino, Kleijnen, and Meloni (2011) consider robust optimization using Taguchi's view of the uncertain world, but replacing his low-order polynomial metamodels by Kriging models; moreover, they use bootstrapping to quantify the variability in the estimated Kriging metamodels. Like Kleijnen et al. (2010) they combine Kriging with Non-Linear Programming (NLP); see Section 6. Changing specific threshold values in the NLP model, they estimate the Pareto frontier. They illustrate the

resulting methodology through a deterministic Economic Order Quantity (EOQ) inventory simulation that has an uncertain input; namely, an uncertain demand rate, and find that robust optimization requires order quantities that differ from the classic EOQ. They also compare their results with their previous results obtained using the low-order polynomials of Response Surface Methodology (RSM) instead of Kriging.

More precisely, Taguchi distinguishes between two types of factors (inputs, variables): (i) *decision* (or control) factors, which managers can control and may be denoted by $\mathbf{d} = (d_1, \ldots, d_k)$; e.g., in inventory management, the order quantity is controllable; and (ii) *environmental* (or noise) factors, which are beyond management's control and which may be denoted by $\mathbf{e} = (e_1, \ldots, e_c)$; e.g., the demand rate in inventory management is an environmental factor.

Taguchi's *statistical* methods are criticized by many statisticians; see the panel discussion in Nair (1992). Dellino et al. use Kriging including Latin Hypercube Sampling (LHS); Kriging is better suited because in computer simulation experiments the experimental area may be much larger than in Taguchi's physical (real-life) experiments so a low-order polynomial may be an inadequate approximation (non-valid metamodel).

Taguchi assumes a single output (say) w and focuses on the ratio of the mean and the variance of this output, also called the signal/noise ratio. Instead of this scalar loss function, Dellino et al. use a NLP model in which $E(w)$ (the mean of the simulation output) is the goal function to be minimized, while s_w (the standard deviation of the goal output) must meet a given constraint:

$$Min_{\mathbf{d}} E(w|\mathbf{d}) \text{ s.t. } s_w \leq T \tag{16}$$

where $E(w|\mathbf{d})$ is the mean over the environmental variables $\mathbf{e}$; this mean can be controlled through $\mathbf{d}$. Dellino et al. use the standard deviation instead of the variance because the standard deviation has the same scale as the mean.

Dellino et al. replace $E(w|\mathbf{d})$ and s_w by their Kriging approximations. Next, they change the threshold T of the constraint in (16), to estimate the Pareto-optimal efficiency frontier; i.e., the mean and standard deviation are the criteria for which a trade-off is required.

Dellino et al. use Kriging, but they are inspired by RSM; RSM is closer to Taguchi's statistical methodology and is developed by Myers, Montgomery, and Anderson-Cook (2009). I summarize the RSM approach to robust optimization in Section 7.1; the reader might wish to skip to Section 7.2.

7.1 RSM for robust optimization: intermezzo

Myers et al. (2009) develop RSM for robust optimization, using the following incomplete second-order polynomial approximation:

$$y = \beta_0 + \boldsymbol{\beta}'\mathbf{d} + \mathbf{d}'\mathbf{B}\mathbf{d} + \boldsymbol{\gamma}'\mathbf{e} + \mathbf{d}'\boldsymbol{\Delta}\mathbf{e} + \epsilon = \boldsymbol{\zeta}'\mathbf{x} + \epsilon \tag{17}$$

where y denotes the regression predictor of $E(w)$, $\boldsymbol{\beta}$ the vector with the k first-order effects of the decision variables $\mathbf{d}$, $\mathbf{B}$ the $k \times k$ symmetric matrix with the purely quadratic effects of $\mathbf{d}$ on the main diagonal and half their interaction effects off the diagonal, $\boldsymbol{\gamma}$ the first-order effects of the environmental factors $\mathbf{e}$, $\boldsymbol{\Delta}$ the decision-by-environmental two-factor interactions, ϵ the NIID residual with $E(\epsilon) = 0$ and constant variance σ_ϵ^2, $\boldsymbol{\zeta}$ is the vector of all regression coefficients, and $\mathbf{x}$ the vector of all regressors. To examine whether (17) is an adequate approximation, Dellino, Kleijnen, and Meloni (2010) use leave-one-out cross-validation.

It is easy to derive that (17) implies

$$E(y) = \beta_0 + \boldsymbol{\beta}'\mathbf{d} + \mathbf{d}'\mathbf{B}\mathbf{d} + \boldsymbol{\gamma}'E(\mathbf{e}) + \mathbf{d}'\boldsymbol{\Delta}E(\mathbf{e}) \tag{18}$$

and

$$var(y) = (\boldsymbol{\gamma}' + \mathbf{d}'\boldsymbol{\Delta})\mathbf{cov}(\mathbf{e})(\boldsymbol{\gamma} + \boldsymbol{\Delta}'\mathbf{d}) + \sigma_\epsilon^2. \tag{19}$$

This mean and variance may be estimated through plugging in the Ordinary Least Squares (OLS) estimates of $\boldsymbol{\zeta}$ in (17) into the right-hand sides of (18) and (19).

Dellino et al. (2010) minimize the estimated mean (say) $\widehat{\overline{y}}$—resulting from plugging the OLS estimates into (18)—while keeping the estimated standard deviation $\widehat{\sigma_y}$—resulting from (19)—below a given Threshold T; see (16). This constrained minimization problem is nonlinear in the decision variables $\mathbf{d}$. Solving this problem through NLP gives $\widehat{\mathbf{d}^+}$, the estimated (see $\widehat{\ }$) robust optimum (see $^+$) for $\mathbf{d}$. Next they vary T, which may give different solutions $\widehat{\mathbf{d}^+}$ with corresponding $\widehat{\overline{y}^+}$ and $\widehat{\sigma_y^+}$. These pairs $(\widehat{\overline{y}^+}, \widehat{\sigma_y^+})$ are then collected to estimate the Pareto frontier. Finally, they estimate the variability of this frontier through *parametric bootstrapping* of $\widehat{\boldsymbol{\zeta}}$—the multivariate-normally distributed OLS estimates—which gave $(\widehat{\overline{y_b}^+}^*, \widehat{\sigma_{y,b}^+}^*)$ $(b = 1, \ldots B)$ with bootstrap sample size B. They illustrate their methodology through EOQ models.

7.2 Kriging for robust optimization

To obtain the I/O simulation data to which *Kriging* models are fitted, simulation analysts often use LHS—a space-filling design (references and websites for other space-filling designs are given by Kleijnen (2008, pp.

127-130)). Dellino et al. (2011) propose the following two approaches using Kriging metamodels:

1. Inspired by Dellino et al. (2010)—see (18) and (19)—they fit two Kriging metamodels; namely, one model for the mean and one for the standard deviation—both estimated from the *simulation* I/O data.

2. Inspired by Lee and Park (2006), Dellino et al. (2011) fit a single Kriging metamodel to a relatively small number (say) n of combinations of the decision variables $\mathbf{d}$ and the environmental variables $\mathbf{e}$. Next they use this metamodel to compute the *Kriging predictions* for the simulation output w for $N >> n$ combinations of $\mathbf{d}$ and $\mathbf{e}$ accounting for the distribution of $\mathbf{e}$.

Sub 1: They select the input combinations for the simulation model through a *crossed* (combined) design for $\mathbf{d}$ and $\mathbf{e}$ (as is also traditional in Taguchian design); i.e., they combine the (say) n_d combinations of $\mathbf{d}$ with the n_e combinations of $\mathbf{e}$ (an alternative is the split-plot design in Dehlendorff, Kulahci, and Andersen (2011)) These n_d combinations are space-filling, to avoid extrapolation. The n_e combinations are *sampled* from their input distribution; they use LHS for this (stratified) sampling. The resulting I/O data enable the following estimators of the n_d conditional means and variances:

$$\overline{w_i} = \frac{\sum_{j=1}^{n_e} w_{ij}}{n_e} \ (i = 1, \ldots, n_d) \tag{20}$$

and

$$s_i^2(w) = \frac{\sum_{j=1}^{n_e} (w_{ij} - \overline{w_i})^2}{n_e - 1} \ (i = 1, \ldots, n_d) \tag{21}$$

where the argument w in $s_i^2(w)$ is added because of the analogous definition in (7), but this argument is suppressed below.

These two estimators are unbiased (no metamodel assumed).

Sub 2: Dellino et al. (2011) select a relatively small n (number of input combinations) using a space-filling design for the $k+c$ input factors ($\mathbf{d}$ and $\mathbf{e}$); i.e., $\mathbf{e}$ is not yet sampled from tits distribution. Next they use these $n \times (k + c)$ simulation input data and their corresponding n outputs w to fit a Kriging metamodel for the output w. Finally, for a much larger design with N combinations they use a space-filling design for $\mathbf{d}$ but for $\mathbf{e}$ they use LHS accounting for the input distribution. They compute the Kriging predictors $\widehat{y}$ for the N outputs. Then they derive the conditional means and standard deviations using (20) and (21) replacing n_e and n_d by N_e and N_d and replacing the simulation

output w by the Kriging predictor $\widehat{y}$ They use these predictions to fit two Kriging metamodels; namely, one Kriging model for the mean and one for the standard deviation of the output.

Sub 1 and 2: The Kriging metamodel combined with the NLP model (16) and varying threshold T gives the estimated Pareto frontier. This frontier, however, is built on estimates of the mean and standard deviation of the simulation output. To quantify the variability in the estimated mean and standard deviation, Dellino et al. (2010) apply *parametric bootstrapping*, whereas Dellino et al. (2011) apply *distribution-free bootstrapping*. Moreover, bootstrapping (both parametric and nonparametric) assumes that the original observations are IID. Because of the crossed design for $\mathbf{d}$ and $\mathbf{e}$, the n_d observations on the output for a given combination of $\mathbf{e}$ are not independent (this dependence may be compared with the dependence created by CRN). Dellino et al. (2011) therefore resample n_e times the n_d-dimensional vectors $\mathbf{w}_j$ $(j = 1, \ldots, n_e)$ with replacement. This resampling gives the n_e bootstrapped observations $\mathbf{w}_j^*$. Then they derive the bootstrapped conditional means $\overline{w_i}^*$ and standard deviations s_i^{2*} using (20) and (21) replacing the simulation output w by the bootstrap observations w^*. To these $\overline{w_i}^*$ and s_i^{2*} they apply Kriging. These Kriging metamodels together with the NLP give the bootstrapped (see superscript $*$) optimal (see $+$) mean $\widehat{\overline{y}}^{+*}$ and standard deviation $\widehat{s}^{+*}$. Repeating this bootstrap sampling B times enables the computation of confidence intervals. Figure 2 is an example of bootstrapped Pareto frontiers based on RSM for the EOQ inventory simulation with mean cost C and standard deviation s_C, which shows that the bootstrapped curves envelop both the original curve and the true curve; see Dellino et al. (2010). An alternative bootstrapped solution is presented in Dellino et al. (2011). Personally, I do not yet know how exactly to use these confidence intervals to account for management's risk attitude. Also see Dellino et al. (2009).

Future research may address the following issues. Instead of minimizing the mean under a variance constraint, we may minimize a specific quantile of the simulation output or minimize the Conditional Value at Risk (CVaR). Other risk measures are the "expected shortfall at level p", which is popular in the actuarial literature. Kriging may be replaced by Generalized Linear Models and NLP by Evolutionary Algorithms. The methodology may also accommodate random simulation models (e.g., (s, S) models), which imply aleatory uncertainty besides epistemic uncertainty.

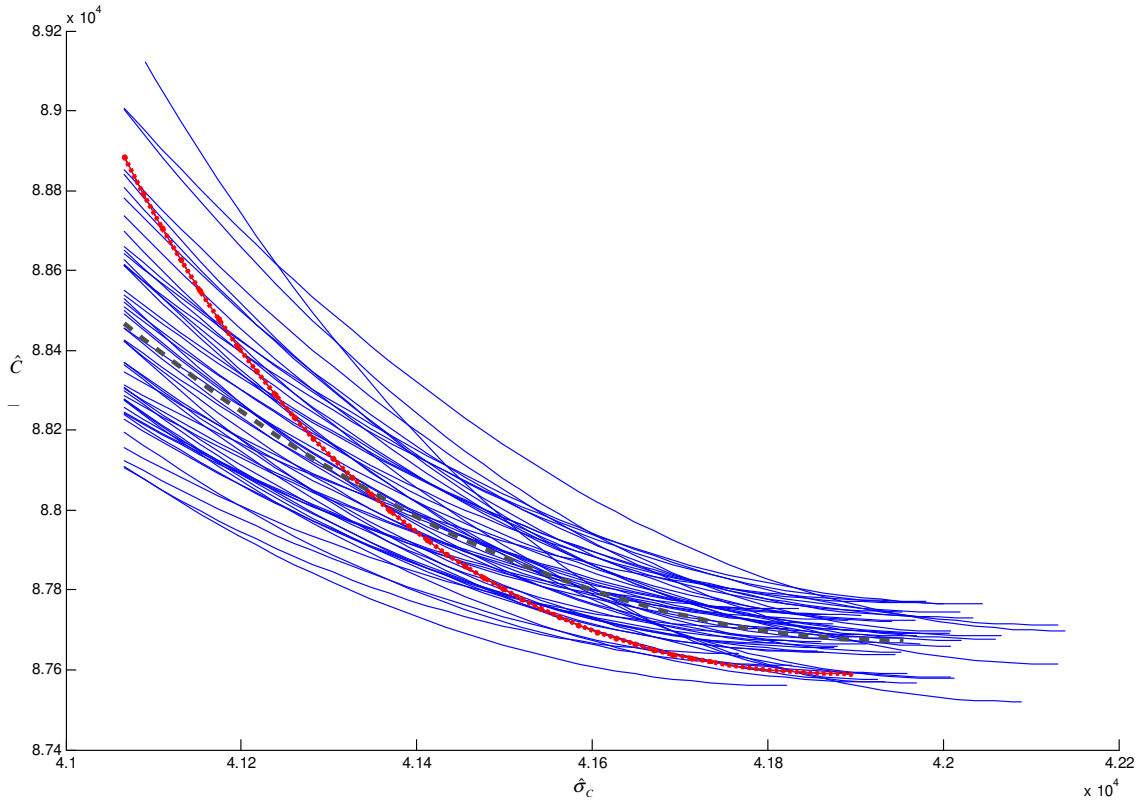

Figure 2: Bootstrapped Pareto frontiers, original estimated frontier (dashed curve), and true frontier (heavy curve)

8 Monotonicity-preserving bootstrapped Kriging

In practice, simulation analysts often know that the I/O function is monotonic; e.g., as the traffic rate increases, so does the mean waiting time; as the EOQ increases, so does the service percentage (or "fill rate") (simulation is still needed to quantify the input effects). To obtain a Kriging metamodel that preserves this characteristic, Kleijnen and Van Beers (2011) use distribution-free bootstrapping assuming each input combination is simulated several times; i.e., replication gives more reliable averaged outputs for the random simulation. Their bootstrapped Kriging gives a noninterpolating metamodel; yet they use standard DACE software. They illustrate their method through a popular single-server simulation model (namely, the M/M/1) with as output either the mean or the 90% quantile of the waiting time distribution; these two outputs are monotonically increasing functions of the traffic rate (and may be nonnormally distributed). Their empirical results demonstrate that their Kriging gives higher probability of covering the true outputs, without lengthening the confidence interval.

More precisely, practical simulation models may be *expensive*. The analysts must then fit a metamodel to a relatively small number n of input combinations $\mathbf{x}_i$ $(i = 1, \ldots, n)$ actually simulated. Unfortunately, these n input combinations replicated m_i times may be so small that the "classic" Kriging metamodel does not preserve the monotonic behavior of the I/O function, but shows *wiggling* (erratic) behavior; bootstrapped Kriging may avoid this wiggling (see Figure 3, further discussed below). Because there is no well-documented software for stochastic Kriging; the authors use the standard DACE software. Their Kriging Their Kriging accounts for *variance heterogeneity* of the simulation outputs; i.e., $var(w_i)$ is an unknown function of $\mathbf{x}_i$.

Monotonically increasing Kriging implies that the estimated *gradients* remain positive as the input increases. This monotonicity preservation implies sensitivity analysis results that are understood and accepted by the clients of the simulation analysts so the decision-makers trust the simulation as a decision support tool. Furthermore, estimated gradients with correct signs may improve simulation optimization.

Figure 3 gives an M/M/1 example is with $m_i = 5$ replicates per point; Kriging uses a Gaussian correlation function. If the analysts require monotonicity for the simulation model's I/O function, then they should obtain so many replicates that the n average simulation outputs also show this property; see again Figure 3; this assumption is realistic if the clients consider the simulation model to be wrong (not valid) if this model generates $\overline{w_i} > \overline{w_{i+1}}$ (higher average waiting time for lower traffic rate). Technically, monotonicity-preserving bootstrap Kriging has

a weaker requirement; namely, $\min_i w_i < \max_i w_{i+1}$.

This bootstrap procedure has the following key steps, assuming no CRN and allowing different numbers of replicates:

1. Resample—with replacement—a replicate number r^* from the uniform distribution defined on the integers $1, \ldots, m_i$; i.e., the uniform density function is $p(r^*) = 1/m_i$ with $r^* = 1, \ldots, m_i$.

2. Replace the r^{th} 'original' output $w_{i;r}$ by the bootstrap output $w^*_{i;r} = w_{i;r*}$.

3. Compute the interpolating Kriging predictor y^* from the bootstrapped I/O data set $(\mathbf{X},\overline{\mathbf{w}}^*)$ where $\mathbf{X}$ denotes the $n \times d$ matrix with the n old combinations of the d simulation inputs and $\overline{\mathbf{w}}^*$ denotes the n-dimensional vector with the bootstrap averages $\overline{w_i}^* = \sum_{r=1}^{m_i} w^*_{i;r}/m_i$ and $i = 1, \ldots, n$. This predictor uses the the MLE (say) $\widehat{\theta}^*$ computed from $(\mathbf{X},\overline{\mathbf{w}}^*_i)$.

4. Keep only the strictly monotonically increasing Kriging predictor y^*; i.e., all d components of the n gradients are positive:

$$\nabla y^*_i > \mathbf{0} \ (i = 1, \ldots, n). \tag{22}$$

   These gradients are provided "for free" by DACE.

This procedure is repeated B times, but it keeps only the (say) B' ($\leq B$) predictors that satisfy $\nabla y^*_{i;b'} > \mathbf{0}$ ($i = 1, \ldots, n$; $b' = 1, \ldots, B'$). For the new input combination (say) $\mathbf{x}_u$, this gives the B' predictions $y^*_{u;b'}$.These $y^*_{u;b'}$ give as the *point estimate* the sample median $y^*_{u;(\lceil 0.50B' \rceil)}$. (Instead of $y^*_{u;(\lceil 0.50B' \rceil)}$ the sample mean $\overline{y^*_u} = \sum y^*_{u;b'}/B'$ might be used—especially when using Kriging for optimization that uses the resulting explicit function.)

These B' Kriging models also give the lower and upper bounds of the (say) 90% *confidence interval*; namely, $y^*_{u;(\lfloor 0.05B' \rfloor)}$ and $y^*_{u;(\lceil 0.95B' \rceil)}$ If this interval turns out to be too wide, then increase B' by increasing the bootstrap sample size B. Confidence intervals in the *classic* Kriging literature assume normality and use the variance estimate $\widehat{\sigma}^2_y$ that ignores the random character of the Kriging (hyper)parameters; see Section 4.

The M/M/1 model is a popular example in random simulation; see Ankenman et al. (2010) and Law (2007, pp. 12-47, 79-83). For the mean and the 90% quantile of the waiting time, the coverages are close to the nominal 90% for the monotonicity-preserving bootstrapped Kriging, whereas classic Kriging gives coverages far below the desired nominal

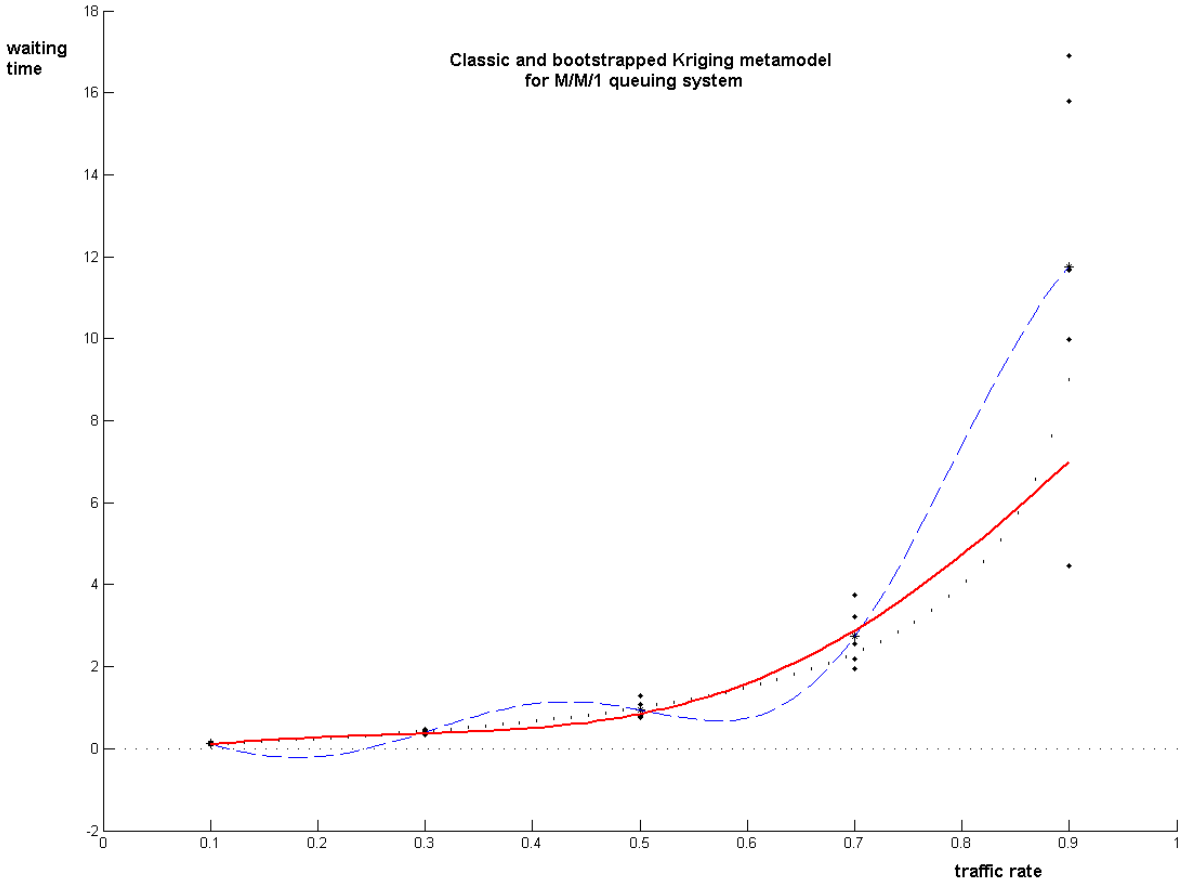

Figure 3: Classic and monotonicity-preserving bootstrapped Kriging metamodels and true I/O function for M/M/1 with $n = 5$ and $m = 5$

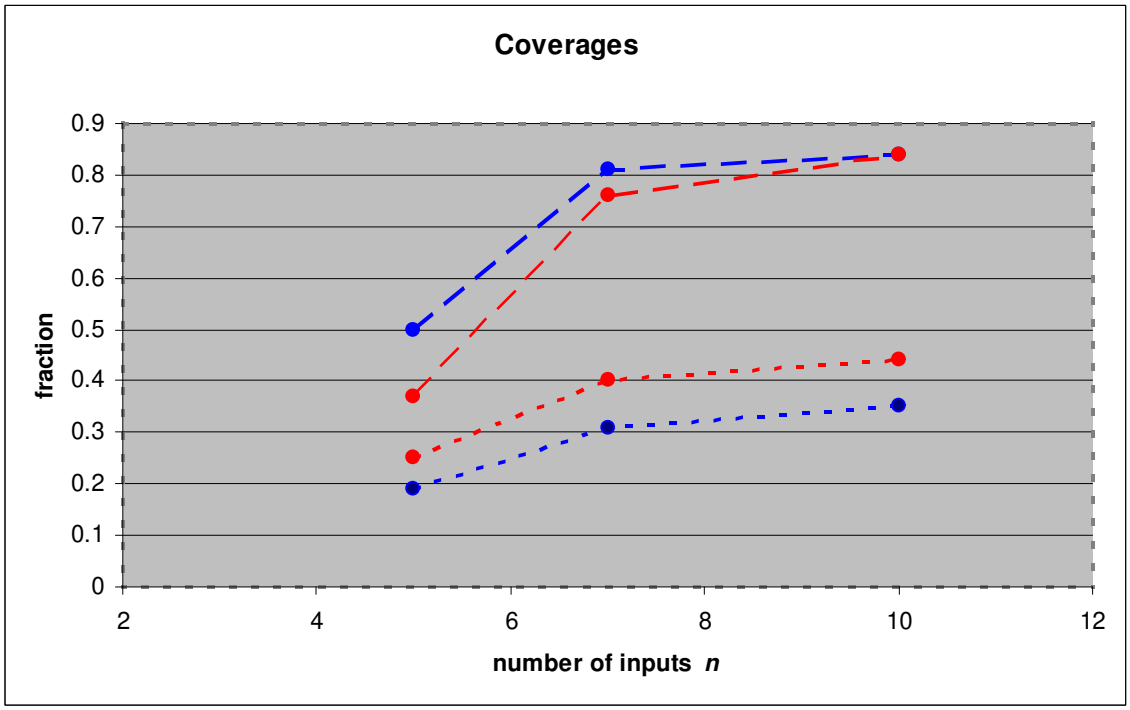

Figure 4: Estimated coverage for mean and quantile in classic and monotonic bootstrapped Kriging versus number of input points n (dashed curves: bootstrap)

value; see Figures 4 and 5. An additional advantage of this bootstrap Kriging is that its confidence interval does not include negative values if negative values are impossible.

Topics for future research may be:

- "Stochastic Kriging" preserving monotonicity
- Preserving other known characteristics; namely, convexity and non-negativeness of the I/O function
- Deterministic simulation with parametric bootstrapping for Kriging preserving known characteristics of the I/O function.

9 Conclusions

I discussed deterministic and random simulators, focusing on optimization via Kriging metamodels. These metamodels may be analyzed through bootstrapping; the various sections demonstrated that the bootstrap is a versatile method but it must be tailored to the specific problem being

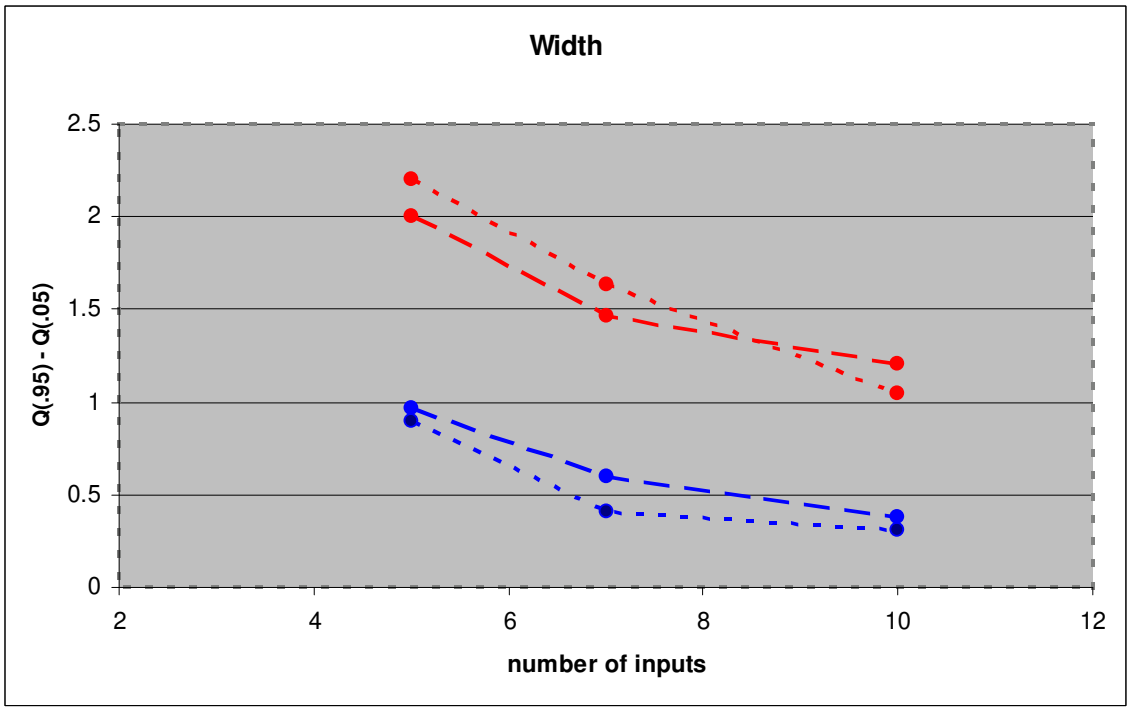

Figure 5: Estimated width of confidence interval for mean and quantile in classic and monotonic bootstrapped Kriging versus number of input points n (dashed curves: bootstrap)

analyzed. Distribution-free bootstrapping is attractive in random simulation that is run several times for the same scenario. A deterministic simulator, however, is run only once for the same scenario, so parametric bootstrapping is used assuming a multivariate Gaussian distribution with (hyper)parameters estimated from the simulator's I/O data.

More specifically, I surveyed:

- EGO with its EI in deterministic simulation, using Kriging with parametric bootstrapping to obtain an unbiased estimator of the Kriging predictor's variance accounting for the randomness resulting from estimating the Kriging parameters.
- Constrained optimization in random simulation, combining Mathematical Programming and Kriging with distribution-free bootstrapping for the validation of the Kriging metamodels.
- Robust optimization accounting for an uncertain environment, combining Kriging metamodels and Mathematical Programming,resulting in a Pareto frontier; the randomness of the Kriging metamodels is analyzed through distribution-free bootstrapping.
- If the I/O function has a specific characteristic such as monotonicity, then this characteristic may be preserved through bootstrapped Kriging.

References

Barton, R.R. and M. Meckesheimer (2006), Metamodel-based simulation optimization. S.G. Henderson, B.L. Nelson, eds. *Handbook in OR & MS 13,* Elsevier. pp. 535-574

Bashyam, S. and M. C. Fu (1998), Optimization of (s, S) inventory systems with random lead times and a service level constraint. *Management Science*, 44, pp. 243–256

Bates, R.A, R.S., Kenett, D.M. Steinberg, and H.P. Wynn (2006), Achieving robust design from computer simulations. *Quality Technology and Quantitative Management*, 3, pp. 161–177

Chen, X., B. Ankenman, and B.L. Nelson (2010), The effects of common random numbers on stochastic Kriging metamodels.Working Paper, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois

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

Dehlendorff, C., M. Kulahci, and K. Andersen (2011), Designing simulation experiments with controllable and uncontrollable factors for applications in health care. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 60, pp. 31-49

Dellino, G. J.P.C. Kleijnen, and C. Meloni (2011), Robust optimization in simulation: Taguchi and Krige combined. *INFORMS Journal on Computing* (accepted)

Dellino, G., J.P.C. Kleijnen, C. Meloni. (2010), Robust optimization in simulation: Taguchi and Response Surface Methodology. *International Journal of Production Economics*, 125, pp. 52-59

Dellino, G., J.P.C. Kleijnen, C. Meloni. (2009), Robust simulation-optimization using metamodels. *Proceedings of the 2009 Winter Simulation Conference*, edited by M.D. Rossini, R.R. Hill, B. Johansson, A. Dunkin, and R.G. Ingalls, pp. 540-550

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

Dykstra, R.L. (1970), Establishing the positive definiteness of the sample covariance matrix. *The Annals of Mathematical Statistics*, 41, no. 6, pp. 2153–2154

Efron, B. and R.J. Tibshirani (1993), *An introduction to the bootstrap*. Chapman & Hall, New York

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

Frazier, P.I. (2011), Learning with Dynamic Programming. In: *Wiley Encyclopedia of Operations Research and Management Science*, Cochran, J.J. , Cox, L.A., Keskinocak, P., Kharoufeh, J.P., Smith, J.C. (eds.), Wiley, New York

Frazier, P., W. Powell, and S. Dayanik (2009), The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing*, 21, pp. 599-613

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

Han G. and T.J. Santner (2008), MATLAB parametric empirical Kriging (MPErK) user's guide. Department of Statistics, The Ohio State University, Columbus, OH 43210-1247

Helton, J.C. and F.J. Davis (2003), Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliability Engineering and System Safety*, 81, pp. 23-69

Janusevskis, J. and R. Le Riche. (2010), Simultaneous kriging-based sampling for optimization and uncertainty propagation. HAL report: hal-00506957

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

Joseph, V. R., Y. Hung, and A. Sudjianto (2008), Blind Kriging: a new method for developing metamodels. *Journal of Mechanical Design*, 130, no. 3, pp. ....

Kelton, W.D., R.P. Sadowski, D.T. Sturrock (2007), *Simulation with Arena*. 4th ed. McGraw-Hill, Boston

Kleijnen, J.P.C. (2007), Risk analysis: frequentist and Bayesians unite! *Proceedings of the 2007 INFORMS Simulation Society Research Workshop*, edited by E. Yucesan, pp. 61-65

Kleijnen, J.P.C. (2008), *Design and analysis of simulation experiments.* Springer, New York

Kleijnen, J.P.C., R.C.H. Cheng, and B. Bettonvil (2001), Validation of trace-driven simulation models: bootstrapped tests. *Management Science*, 47, no. 11, pp.1533-1538

Kleijnen, J.P.C. and D. Deflandre (2006), Validation of regression metamodels in simulation: bootstrap approach.*European Journal of Operational Research*, 170, pp. 120–131

Kleijnen, J.P.C. and W.C.M. Van Beers (2011), Monotonicity-preserving bootstrapped Kriging metamodels for expensive simulations. Working Paper

Kleijnen, J.P.C., W.C.M. Van Beers, and I. Van Nieuwenhuyse (2010), Constrained optimization in simulation: a novel approach. *European Journal of Operational Research*, 202, pp. 164-174

Kleijnen, J.P.C., W.C.M. Van Beers, and I. Van Nieuwenhuyse (2011), Expected improvement in efficient global optimization through bootstrapped Kriging. Working Paper

Kleijnen, J.P.C. and W. van Groenendaal (2000), Measuring the quality of publications: new methodology and case study. *Information Processing & Management,* 36, pp. 551-570

Law, A.M. (2007), *Simulation modeling and analysis; fourth edition.* McGraw-Hill, Boston

Lee, K.H. and G.J. Park (2006), A global robust optimization using Kriging based approximation model. *Journal of the Japanese Society of Mechanical Engineering*, 49, pp. 779-788

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

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

Myers, R.H., D.C. Montgomery, and C.M. Anderson-Cook (2009), *Response surface methodology: process and product optimization using designed experiments; third edition.* Wiley, New York

Nair, V.N., editor.(1992), Taguchi's parameter design: a panel discussion. *Technometrics*, 34, pp. 127-161

Novikov, I. and B. Oberman (2007), Optimization of large simulations using statistical software *Computational Statistics & Data Analysis*, 51, no. 5, pp. 2747-2752

Picheny, V., D. Ginsbourger, and Y. Richet (2010), Noisy Expected Improvement and on-line computation time allocation for the optimization of simulators with tunable fidelity. In: *2nd International Conference on Engineering Optimization*, 6-9 September 2010, Lisbon, Portugal

Rasmussen, C.E. and C. Williams (2006), *Gaussian processes for machine learning*, The MIT Press, Cambridge, Massachusetts

Regis, R.G. (2011), Stochastic radial basis function algorithms for large-scale optimization involving expensive black-box objective and constraint functions. *Computers & Operations Research*, 38, pp. 837-853

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

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

Stein, M.L. (1999), *Statistical interpolation of spatial data: some theory for Kriging,* Springer

Taguchi, G. (1987), *System of experimental designs, volumes 1 and 2*. UNIPUB/ Krauss International, White Plains, New York

Van Beers, W.C.M. and J.P.C. Kleijnen (2008), Customized sequential designs for random simulation experiments: Kriging metamodeling and bootstrapping. *European Journal of Operational Research*, 186, no. 3, pp.1099–1113

Villemonteix, J., E. Vazquez, and E. Walter (2009), An informational approach to the global optimization of expensive-to-evaluate functions. Journal of Global Optimization,44, no. 4, pp. 509-534

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

Acknowledgement 1 *I thank Bertrand Iooss (Electricité de France) for inviting me to present a seminar at the conference "Stochastic and noisy simulators" organized by the French Research Group on Stochastic Analysis Methods for COdes and NUMerical treatments called "GDR MASCOT-NUM" in Paris on May 17, 2011.*