Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, eds.

CONVEX AND MONOTONIC BOOTSTRAPPED KRIGING

Jack P.C. Kleijnen
Ehsan Mehdad

Tilburg University
Warandelaan 2
5037 AB Tilburg, NETHERLANDS

Wim C.M. van Beers

University of Amsterdam
Roetersstraat 11
1018 WB Amsterdam, NETHERLANDS

ABSTRACT

Distribution-free bootstrapping of the replicated responses of a given discrete-event simulation model gives bootstrapped Kriging (Gaussian process) metamodels; we require these metamodels to be either convex or monotonic. To illustrate monotonic Kriging, we use an M/M/1 queueing simulation with as output either the mean or the 90% quantile of the transient-state waiting times, and as input the traffic rate. In this example, monotonic bootstrapped Kriging enables better sensitivity analysis than classic Kriging; i.e., bootstrapping gives lower MSE and confidence intervals with higher coverage and the same length. To illustrate convex Kriging, we start with simulation-optimization of an (s, S) inventory model, but we next switch to a Monte Carlo experiment with a second-order polynomial inspired by this inventory simulation. We could not find truly convex Kriging metamodels, either classic or bootstrapped; nevertheless, our bootstrapped "nearly convex" Kriging does give a confidence interval for the optimal input combination.

1 INTRODUCTION

Many realistic simulation models have known characteristics such as *convexity* and *monotonicity*. For example, simulation models of supply chains consist of a sequence of submodels (building blocks, modules) for queues and inventories; higher traffic rates monotonically increase mean waiting time, and reorder levels and order quantities are often assumed to have a unique optimal combination because the cost function is convex (instead of having multiple local optima). However, in their classic textbook on convex optimization Boyd and Vandenberghe (2004) study problems with explicit functions, whereas simulation problems have implicit functions that are determined by the underlying simulation model. In this paper, we use a metamodel to approximate such an implicit function (also see Nesterov 2003, pp. 171-172).

Metamodels (also called response surfaces, emulators, etc.) serve sensitivity analysis of the simulation models and optimization of the simulated systems. There are several types of metamodels, but the most popular types are linear regression analysis and Kriging (or Gaussian process) models; many references to various types of metamodels are given by Kleijnen 2008, p. 8. Well-known types of monotonic regression models are isotonic regression and "rank" regression; see Kleijnen 2008, pp. 98, 162. We, however, focus on *Kriging*. Monotonic Kriging metamodels are also examined by Kleijnen and van Beers (2011); we summarize and update that publication, and extend it to convexity.

To estimate the Kriging metamodel, we simulate (say) n combinations (or points) $\mathbf{x}_i$ of the $k \geq 1$ simulation inputs; we replicate these combinations m_i times ($i = 1, ..., n$) We assume that the simulation model is *expensive*; i.e., the simulation requires much computer time to obtain the outputs $w_{i;r}$ ($r = 1, ..., m_i$), so the set of input/output (I/O) data may be so small that "classic" Kriging does not preserve the assumed characteristic, and shows *wiggling* (erratic) behavior. We therefore derive *bootstrapped Kriging* that is meant to avoid this wiggling. Bootstrapping is discussed in the classic textbook by Efron and Tibshirani (1993); additional recent references are given by Kleijnen 2008, pp. 81.

978-1-4673-4781-5/12/$31.00 ©2012 IEEE

More specifically, *classic* Kriging is an *exact interpolator*; i.e., the Kriging predictions $y(\mathbf{x}_i) = y_i$ equal the simulation outputs $w(\mathbf{x}_i) = w_i$ for the n "old" (actually simulated) input combinations $\mathbf{x}_i$. This Kriging is often applied in deterministic simulation, which is popular in engineering. In stochastic simulation, however, this interpolation property is not desirable, because this simulation gives different outputs at the same $\mathbf{x}_i$ whenever the pseudo-random number (PRN) seed changes. The Kriging metamodel may be slightly changed such that it does not interpolate the n averaged outputs $\overline{w}_i = \sum_{r=1}^{m_i} w_{i;r}/m_i$; see Ankenman, Nelson, and Staum (2010). We use the free MATLAB Kriging toolbox called *DACE*, which is well documented by Lophaven, Nielsen, and Sondergaard (2002); DACE is often applied in practice (alternative software is mentioned in Section 5).

To obtain Kriging metamodels that are either convex or monotonic, we apply *distribution-free bootstrapping* to the old simulation I/O data; i.e., we resample—with replacement—the m_i replicated simulation outputs $w_{i;r}$. This bootstrapping is computationally inexpensive compared with the computer time required by expensive simulation. These bootstrapped Kriging metamodels imply sensitivity analysis and optimization results that are understood and accepted by the users so they have more confidence in the underlying simulation model as part of the decision support system (DSS). We investigate whether our monotonic Kriging gives "better" predictions than classic Kriging does; i.e., we compare the mean squared error (MSE)—which is the standard criterion in Kriging—and the coverage and width of the confidence intervals (CIs) for the Kriging predictions. We also examine convex Kriging metamodels, expecting that these metamodels give better estimates of the optimal input combination.

To illustrate our method and estimate its performance, we use the two submodels that are most often used in simulation; namely, the single-server ($GI/G/1$) queuing model and the (s, S) inventory model; see the various textbooks on simulation including Kroese, Taimre, and Botev 2011, pp. 287-292. We use Kroese, Taimre, and Botev (2011) because we prefer MATLAB code and this book has a web page with MATLAB code for these models; namely, Kroese, D. P. (2012).

Our main conclusions—for simulations that are so expensive that sample sizes are so small that classic Kriging gives wiggling behavior—will be: (i) Bootstrapped monotonic Kriging gives smaller estimated MSE, albeit not significantly smaller; it also gives CIs with higher coverage and acceptable length; (ii) bootstrapped convex Kriging gives confidence intervals for the values of the optimal input combination.

Note: If there would be no replicates ($m_i = 1$) (as in deterministic simulation), then our distribution-free bootstrapping would not apply and we would resort to parametric bootstrapping assuming a Gaussian process with parameters estimated from the simulation I/O data.

The remainder of our paper is organized as follows. Section 2 summarizes classic Kriging, and details our bootstrapped Kriging preserving the assumed characteristic (convexity or monotonicity). Section 3 details monotonic bootstrapped Kriging illustrated through the M/M/1 simulation model. Section 4 details convex bootstrapped Kriging illustrated through an (s, S) simulation model and an artificial examples inspired by this inventory simulation. Section 5 presents conclusions and topics for further research.

2 BOOTSTRAPPED KRIGING WITH PRESERVED CHARACTERISTICS

First we summarize the basics of *classic Kriging* as follows. Kriging uses the $n \times n$ matrix $\mathbf{\Gamma} = [cov(w_i, w_{i'})]$ with $i, i' = 1, \ldots, n$ and the n-dimensional vector $\gamma = [cov(w_i, w_0)]$ where w_i denotes the output of $\mathbf{x}_i$ (an old input combination already simulated), w_0 denotes the output of $\mathbf{x}_0$, the combination to be predicted—which may be either new or old. These $\mathbf{\Gamma}$ and γ often use the Gaussian correlation function $\mathcal{R}(\theta, \mathbf{x}_i, \mathbf{x}_{i'}) = \Pi_{j=1}^{k} \exp[-\theta_j h_j^2]$ with $h_j = \left|x_{i;j} - x_{i';j}\right|$ and θ_j measuring the importance of input j (Kriging in simulation implies that each of the k inputs is measured on a quantitative scale such that the Euclidean distance h is defined). To estimate the unknown Kriging parameters, Kriging usually applies maximum likelihood estimation (MLE); the resulting MLE estimators are denoted by a hat (e.g., $\widehat{\gamma}$, $\widehat{\mathbf{\Gamma}}$, $\widehat{\mu}$, $\widehat{\theta}_j$). The predictor for point $\mathbf{x}_0$ is

$$\widehat{\widehat{y(\mathbf{x}_0)}} = \widehat{\mu} + \widehat{\gamma}^T \widehat{\mathbf{\Gamma}}^{-1} (\mathbf{w} - \widehat{\mu}\mathbf{1}) \tag{1}$$

with $\widehat{\mu} = (\mathbf{1}^T\widehat{\mathbf{\Gamma}}^{-1}\mathbf{1})^{-1}\mathbf{1}^T\widehat{\mathbf{\Gamma}}^{-1}\mathbf{w}$ and $\mathbf{w} = (w_1, \ldots, w_n)^T$; we use a "double hat" $\widehat{\widehat{y}}$ to emphasize that this predictor uses parameters estimated through MLE (obviously, this predictor is nonlinear).

The predictor (1) implies the following gradient with respect to $\mathbf{x}$ at the point $\mathbf{x}_0$:

$$\widehat{\nabla y(\mathbf{x}_0)} = \mathbf{J}_\gamma^T\widehat{\mathbf{\Gamma}}^{-1}(\mathbf{w} - \widehat{\mu}\mathbf{1}) \tag{2}$$

where $\mathbf{J}_\gamma$ is the Jacobian of $\widehat{\gamma}$ so $\mathbf{J}_\gamma = \widehat{\nabla\gamma(\mathbf{x}_0)}$. This gradient is provided by DACE; see Lophaven, Nielsen, and Sondergaard 2002, pp. 16-18 (and also Exercise 5.5 in Kleijnen 2008, p. 143).

Classic Kriging also gives CIs; see Lophaven, Nielsen, and Sondergaard 2002, p. 4 and Santner, Williams, and Notz 2003, p. 96. These CIs assume normality and uses $\widehat{\sigma}^2_{\widehat{y}}$ which estimates the variance of the classic predictor $\widehat{y}$ ignoring the random character of the Kriging weights resulting from estimating the Kriging parameters.

Most designs for Kriging in simulation use Latin Hypercube Sampling (LHS), which implies that each of the k inputs has n distinct values that are either exactly or approximately equally spaced; see Kleijnen 2008, pp. 126-130.

Next we summarize our bootstrapped Kriging. Distribution-free bootstrapping assumes that all n old points are replicated "enough" times: $m_i \gg 2$; e.g. m_i = 5 in the M/M/1 example in Figure 1 (further discussed below). This bootstrap gives the bootstrapped observations $w^*_{i;r}$ with r = 1, ..., m_i; bootstrapping uses the same sample size m_i as the original simulation. These m_i bootstrapped simulation outputs give the bootstrapped average $\overline{w^*_i}$. At different points $\mathbf{x}_i$, the simulation outputs $w_{i;r}$ have different means and variances so they are not independently and identically distributed (IID). So the vector of bootstrapped average simulation outputs is $\overline{\mathbf{w}^*} = (\overline{w^*_i}, \ldots, \overline{w^*_n})^T$.

We repeat this bootstrapping (say) B times; B is called the "bootstrap sample size". A typical choice is B = 100—but after observing the results for B bootstrap samples, we might select more samples if necessary; e.g., we double B. So we obtain B bootstrapped Kriging predictors $\widehat{\widehat{y^*_b}}$ with $b = 1, \ldots, B$; this $\widehat{\widehat{y^*_b}}$ uses the MLE computed from $(\mathbf{X}, \overline{\mathbf{w}^*_b})$. From these B bootstrapped predictors we accept the (say) B_a ($\leq B$) predictors that satisfy the required characteristic (convexity or monotonicity) and reject the remaining predictors; we select B_a such that these accepted predictors give reasonable CIs. Our B_a accepted bootstrapped Kriging predictors $\widehat{\widehat{y^*_{b_a}}}$ (b_a = 1, ..., B_a) are not exact interpolators of $\overline{\mathbf{w}_i}$ (these $\widehat{\widehat{y^*_{b_a}}}$ are exact interpolators of $\overline{\mathbf{w}^*_{b_a}}$ because we compute these predictors through DACE).

Altogether our bootstrapped convex or monotonic Kriging procedure runs as follows.

1. Read the simulation I/O data $(\mathbf{X}, \mathbf{w}_i)$ with $\mathbf{w}_i = (w_{i;1}, \ldots, w_{i;m_i})$, the bootstrap sample size B, and the number of predictors to be accepted B_a.
2. Initialize the accepted number of bootstrapped Kriging models b_a = 0; the bootstrap sample number b = 1.
3. Initialize the simulation input combination i = 1; the replicate number $r = 1$.
4. Resample—with replacement—a replicate number r^* from $U(1, m_i)$, which denotes the uniform distribution defined on the integers $1, \ldots, m_i$.
5. Replace the "original" output $w_{i;r}$ by the bootstrap output $w^*_{i;r} = w_{i;r*}$.
6. If $r < m_i$ then $r = r + 1$ and return to Step 4 else proceed to the next step.
7. If $i < n$ then $i = i + 1$ and return to Step 4; else proceed to the next step.
8. Compute the interpolating bootstrapped Kriging predictor $\widehat{\widehat{y^*}}$ (short-hand notation y^*) from the bootstrapped I/O data set $(\mathbf{X}, \overline{\mathbf{w}^*})$ where $\mathbf{X}$ denotes the $n \times k$ matrix with the n old combinations of the k 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 $i = 1, \ldots, n$ (so $y^*_i = \overline{w^*_i}$); compute this predictor for all old points and selected new points.
9. If $\widehat{\widehat{y^*_i}}$ (the bootstrapped predictor of Step 8) is accepted, then $b_a = b_a + 1$.

10. If $b < B$ then $b = b + 1$; return to Step 3; else proceed to the next step.
11. If $b_a < B_a$ then B = $2B$; return to Step 3; else proceed to the next step.
12. Compute point estimates and CIs from the B_a accepted Kriging metamodels.

3 MONOTONICITY: M/M/1 QUEUE SIMULATION

There are several variants of the $GI/G/1$ model. In academia, the most popular variant is the M/M/1 model; i.e., the interarrival distribution GI becomes exponential with rate λ (or mean $1/\lambda$) denoted as $\text{Exp}(\lambda)$, and the service distribution becomes $\text{Exp}(\mu)$ (so the model becomes Markovian). Implicitly, the queuing discipline is first-in-first-out (FIFO), the waiting room has infinite capacity, customers do neither balk nor renege, etc. The input is the traffic rate $x = \rho = \lambda/\mu$, which is assumed to be smaller than 1 so that the steady state can be reached. We study two outputs: (i) the steady-state mean waiting time μ_w; (ii) the steady-state 90% quantile $w_{.90}$ defined by $P(w_t \leq w_{.90}|t \to \infty) = 0.9$. The classic estimator of μ_w is the time-series average $\overline{w} = \sum_{t=1}^{T} w_t/T$; the estimator of $w_{.90}$ is $\widehat{w_{.90}} = w_{(\lceil .90T \rceil)}$ (the subscript () denotes order statistics). To verify the simulation results, we use Kleijnen and van Beers (2011)'s analytical results: $w_{.90} = -\ln(0.1/x)/\mu(1-x)$ and $\mu_w = x/[\mu(1-x)]$.

To estimate the sampling variability of $\overline{w}$ and $\widehat{w_{.90}}$, we use $m \geq 2$ *replicates* (each of length T); replicate r (r = 1, ..., m) gives $\overline{w_r}$ and $\widehat{w_{.9;r}}$. Kleijnen and van Beers (2011) find that $\overline{w_r}$ and $\widehat{w_{.9;r}}$ are not normally distributed if the simulation run is as short as T = 1000, even for the relatively low traffic rate 0.5.

We assume that n and m_i are so small that the fitted Kriging metamodel may be non-monotonic; Kleijnen and van Beers (2011) give the example in Figure 1. We assume that we do obtain so many replicates that the n average simulation outputs are increasing monotonically; see again Figure 1. This assumption is realistic if otherwise the users consider the simulation model to be wrong (not valid if an average simulated waiting time is higher for a lower traffic rate). Technically, monotonic bootstrap Kriging has a weaker requirement; namely, $\min_i w_i < \max_i w_{i+1}$ with $\rho_i > \rho_{i-1}$; see Kleijnen and van Beers (2011).

The B_a (accepted) monotonically increasing bootstrapped Kriging metamodels imply that the gradients at the n old points are positive:

$$\frac{dy^*_{i;b_a}}{dx_i} > 0 \ (i = 1, \ldots, n) \ (b_a = 1, \ldots, B_a). \tag{3}$$

Wiggling may also occur at new points, so we check (say) 100 new points spread uniformly across the experimental range.

From the B_a accepted predictors we compute predictions y^*_u for v new input combinations x_u (u = 1, ..., v), which form a *test set*; the same Kriging metamodel is used to predict the outputs for the v different test points. Using these B_a predictions for point u, our *point estimate* is the sample median $y^*_{u;(\lceil 0.50B_a \rceil)}$. Besides this point estimate, we also compute the following simple *90% CI; namely,* $(y^*_{u;(\lfloor 0.05B_a \rfloor)}, y^*_{u;(\lceil 0.95B_a \rceil)})$ (more complicated CIs are discussed in Efron and Tibshirani (1993)). If this interval turns out to be too wide, then we increase B_a by increasing the bootstrap sample size B; e.g., in our M/M/1 example we start with $B = 100$ but augment B with another 100 until either $B_a \geq 100$ or (to avoid excessive computational time) $B = 1000$. It turns out that only in 5 of the 100 "macro-replicate" (which differ only in their PRN seeds), B = 100 gives only $B_a < 100$ monotonic Kriging models, so another 100 bootstrap samples are generated. These B_a bootstrap samples enable the estimation of both the *coverage* and the *width* of the CIs for bootstrapped and classic Kriging—averaged over all ν test points. Actually, v = 25 new points are selected—through LHS—such that no extrapolation is needed (Kriging is believed to give a poor extrapolator).

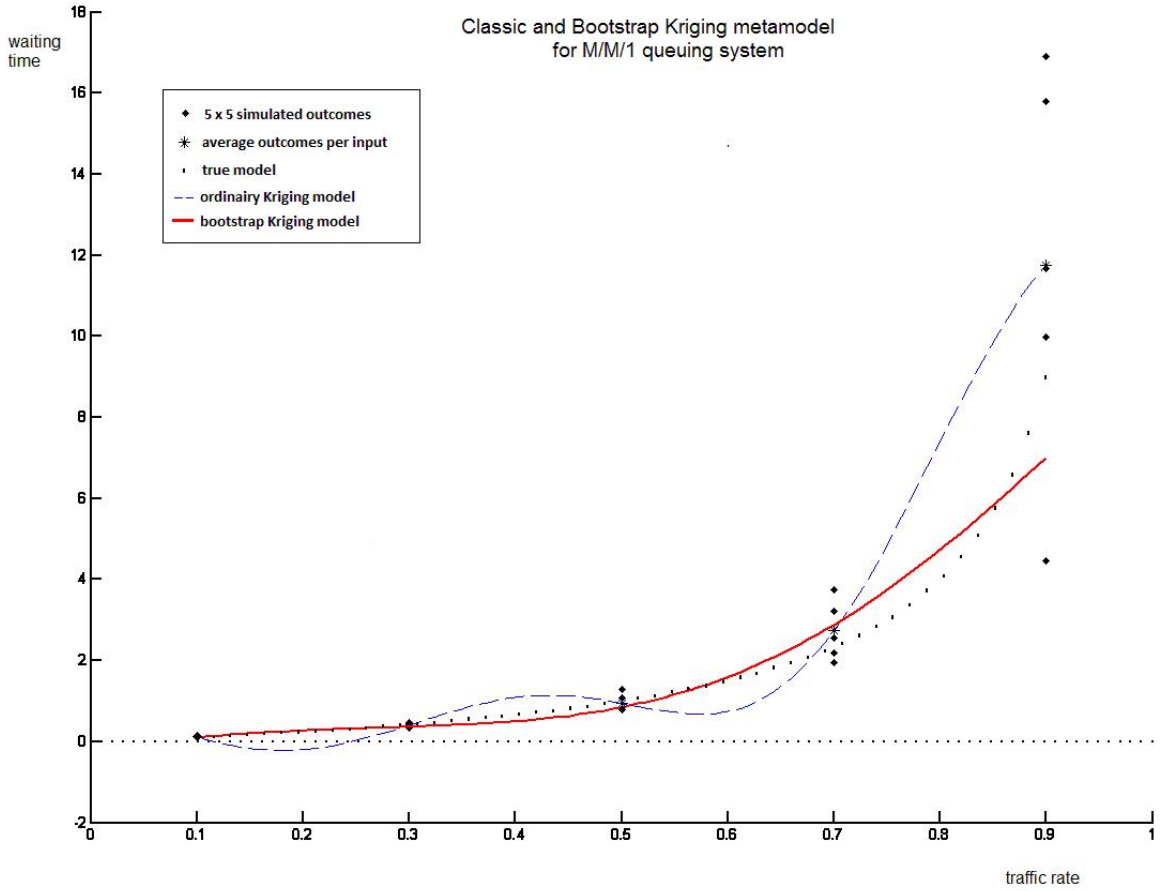

Figure 1: Classic Kriging and monotonic bootstrapped Kriging, and true I/O function for M/M/1 with n = 5, m = 5, T =1000.

To estimate whether the bootstrapped median point predictor for the true output (say) ζ is better than the classic Kriging predictor, the *Integrated MSE* (IMSE) is estimated:

$$\widehat{IMSE}^* = \frac{\sum_{u=1}^{v}(y^*_{u;(\lceil 0.50B_a \rceil)} - \zeta_u)^2}{v}; \widehat{IMSE} = \frac{\sum_{u=1}^{v}(y_u - \zeta_u)^2}{v}. \quad (4)$$

Estimating the coverage of the bootstrapped CIs uses the indicator function $I^*_u = 1$ if $y^*_{u;(\lfloor 0.05B_a \rfloor)} < \zeta_u < y^*_{u;(\lceil 0.95B_a \rceil)}$; else $I^* = 0$. The classic Kriging uses the classic estimated predictor variance $\widehat{\sigma^2_{\widehat{y}_u}}$ (ignoring the randomness of the estimated Kriging parameters) so $I_u = 1$ if $\widehat{\widehat{y}_u} - 1.64\widehat{\sigma}_{\widehat{y}_u} < \zeta_u < \widehat{\widehat{y}_u} + 1.64\widehat{\sigma}_{\widehat{y}_u}$; else $I = 0$. This formula shows that the classic CI is symmetric around its point estimate and may include negative values—even if negative waiting times are impossible. Analogously to the IMSE defined in (4), these indicator functions are averaged over all v test points: $\overline{I^*} = \sum_{u=1}^{v} I^*_u/v$; $\overline{I} = \sum_{u=1}^{v} I_u/v$. Let $\overline{I^*}$ and $\overline{I}$ in macro-replicate l be denoted by $\overline{I^*_l}$ and $\overline{I_l}$ with l = 1, ..., L; e.g., L = 100. Bootstrapping then gives better *coverage* if $\overline{\overline{I^*}} = \sum_l \overline{I^*_l}/L$ is closer to the nominal value 0.90 than $\overline{\overline{I}} = \sum_l \overline{I_l}/L$.

These L macro-replicates also give a 90% CI for the IMSE in classic Kriging; namely, $\overline{\widehat{IMSE}} \pm 1.64 s(\widehat{IMSE})/L^{1/2}$ where $\overline{\widehat{IMSE}} = \sum_{l=1}^{L} \widehat{IMSE}_l/L$ and $s(\widehat{IMSE}) = [\sum_{l=1}^{L}(\widehat{IMSE}_l - \overline{\widehat{IMSE}})^2/(L-1)]^{1/2}$. For bootstrapped Kriging, analogous formulas apply. For the coverage and the length of the CI also analogous formulas apply.

Kleijnen and van Beers (2011) give the estimated IMSE for the average and the 90% quantile. Bootstrapping gives smaller estimated IMSE, albeit not significantly smaller (as expected, the 90% quantile has larger IMSEs than the mean has). Bootstrapping gives significantly higher estimated coverages for the mean and the quantile. Unfortunately, all estimated coverages are significantly lower than the nominal (prescribed) value 90%. Bootstrapping gives average widths that are not significantly shorter. The variability of the width is smaller for bootstrapped Kriging. Altogether, bootstrapping gives better coverage without lengthening the CI.

To further examine this low coverage, Kleijnen and van Beers (2011) increases n from 5 to 10. This change increases the estimated coverages for both classic and monotonic Kriging; this improved coverage may be explained by the better fit of the Kriging model resulting from an "adequate" sample size; also see Loeppky, Sacks, and Welch (2009), suggesting that a valid Kriging metamodel requires n = $10k$ (which in the M/M/1 example implies n = 10). *These coverages are close to the nominal 90% for monotonic bootstrapped Kriging, whereas classic Kriging still gives coverages far below the desired nominal value.* This improved coverage does not require significantly longer CIs.

4 CONVEXITY: (S, S) INVENTORY SIMULATION

A general textbook on convex optimization is Boyd and Vandenberghe (2004). We focus on (s, S) inventory models. There are many variants of this (s, S) model, but we wish to select a model with a convex I/O function. We therefore exclude models with a service-rate constraint; such a constraint would imply two outputs—namely, the service rate and the sum of ordering cost and holding cost. More specifically, we select Kroese, Taimre, and Botev (2011)'s model:

$$C(s, S) = c_1 S + c_2 f_{neg} + c_3 f_{ord} \quad (5)$$

with total costs $C(s, S)$, holding cost $c_1 S$, backorder cost $c_2 f_{neg}$ where f_{neg} denotes the fraction of time with negative net-inventory, and ordering cost $c_3 f_{ord}$ where f_{ord} denotes the frequency of orders; obviously, $s \geq 0$ and $S \geq s$. Kroese, Taimre, and Botev (2011) select the parameter values c_1 = 5, c_2 = 500, and c_3 = 100. Furthermore, for the distributions of the interarrival time, demand size, and lead time they select $\text{Exp}(1/5)$, $U(0, 10)$, and $U(5, 10)$. They run the simulation during T = 1000 days. Through the cross-entropy method they find the estimated optimum $(\widehat{s_{opt}}, \widehat{S_{opt}})$ = (15.56, 19.42) with estimated minimum cost $\widehat{C_{opt}}$ = 149.6.

Actually, the crucial question is whether the specified $(s,\ S)$ inventory simulation model implies a convex I/O function (the inputs s and S—besides the distributions of the interarrival time, demand size, and lead time—implicitly determine the probability functions of the random variables in (5); namely, f_{neg} and f_{ord}). To answer this question, we proceed as follows.

Like Kroese, Taimre, and Botev (2011), we fix the simulation run length at T = 1000. We select an experimental area that ranges from the minimum to the maximum of the reorder level s that we think to be reasonable—given the demand and lead time; i.e., we select $0 \leq s \leq 100$. Analogously, we select $0 \leq Q \leq 100$ with $Q = S - s$. (Originally, we selected a much smaller area centered around Kroese, Taimre, and Botev (2011)'s optimum solution, but this area implied a low signal-noise ratio so it was hard to fit a Kriging metamodel.) Within this area ($0 \leq s, Q \leq 100$) we select n = 20 combinations of (s, Q), because of Loeppky, Sacks, and Welch (2009)'s rule-of-thumb ($n = 10k$). To select the specific n combinations, we use popular LHS. To obtain reliable simulation responses, we first obtained a pilot-sample of m = 10 replicates for each of these n combinations, and found that the signal-noise ratio was rather low; so we decide to obtain m = 5000 replicates per combination. This gives the average simulated output per combination $\overline{C_i} = \sum_{r=1}^{m} C_{i;r}/m$ and its standard error $\widehat{\sigma}_i = \{\sum_{r=1}^{m}[C_{i;r} - \overline{C_i})]^2/[(m_i - 1)m_i]\}^{1/2}$ so the signal-noise ratio is $\overline{C_i}/\widehat{\sigma}_i$ (i = 1,..., n). This simulation experiment gives the I/O data of Table 1; the last column will be explained after (6) (this table does not display the individual outputs $C_{i;r}$, which we shall bootstrap to find $C^*_{i;r}$). This table and its plot (which we do not display) suggest that the simulation's I/O function is convex in the subarea with relatively low s and Q (which includes Kroese, Taimre, and Botev (2011)'s optimum); our formal analysis proceeds as follows. The second-order conditions (see Boyd and

Table 1: I/O data of (s, S) simulation with $0 \leq s \leq 100$ and $0 \leq\ Q \leq 100$ ($Q = S - s$).

i	s_i	Q_i	$\overline{C_i}$	σ_i	PSD?
1	60.7090	94.3850	776.4399	0.0872	NO
2	65.7360	6.2017	370.3396	0.8975	NO
3	28.1440	35.1720	319.8780	1.2138	YES
4	31.1940	57.9680	447.6184	0.6270	NO
5	54.9630	77.5910	663.9453	0.1090	NO
6	91.8080	13.2570	531.2722	0.4961	NO
7	1.4142	40.1580	282.4015	8.8624	YES
8	17.5980	82.1300	502.6232	2.4031	NO
9	42.2870	70.2050	563.7797	0.1775	YES
10	87.1490	46.7590	671.4780	0.1668	NO
11	12.1530	67.2550	407.8239	4.4803	NO
12	79.9920	96.8710	885.2565	0.0859	NO
13	74.3760	31.3110	531.2612	0.2362	NO
14	35.5270	26.0240	311.2771	0.6009	YES
15	57.1530	64.7110	610.7371	0.1239	YES
16	96.3610	86.7030	916.3779	0.0940	NO
17	20.1100	51.2040	361.1883	2.4102	YES
18	7.3929	16.6630	195.6253	13.4523	YES
19	49.9980	20.8630	358.3725	0.3535	YES
20	81.0160	2.1276	431.7596	1.2542	NO

Vandenberghe 2004, p. 71) imply that if a *convex* function (say) f is twice differentiable, then this f has a *Hessian* that is positive semi-definite (PSD). To verify whether the inventory simulation has indeed a convex I/O function $E[C(s, S)]$ with $C(s, S)$ defined in (5), we fit a Kriging metamodel $\widehat{\widehat{C}}$ (see (1)) to the I/O data in Table 1; i.e., we apply DACE to the averages $\overline{C_i}$. This Kriging model implies estimates $\partial\widehat{\widehat{C}}/\partial s$ and $\partial\widehat{\widehat{C}}/\partial Q$ at a specific point (say) (s_0, S_0); see again (2). This metamodel is less precise at interpolated new points than it is at simulated old points; nevertheless, we compute not only the predicted first-order derivatives at the n old points, but also at 10000 new points on a 100×100 grid (in their M/M/1 simulation Kleijnen and van Beers (2011) also estimate derivatives at new points). These first-order derivatives imply the following estimates of the second-order derivatives at the point (s_0, Q_0), given the old points i (i =1,

..., n):

$$\kappa_{s;i} = \left.\frac{\partial^2 \widehat{\gamma}_i}{\partial s^2}\right|_{s_0;Q_0} = [-2\widehat{\theta_s} + 4\widehat{\theta_s}^2(s_0 - s_i)^2]\exp[-\widehat{\theta_s}(s_0 - s_i)^2 - \widehat{\theta_Q}(Q_0 - Q_i)^2]$$

$$\kappa_{s;Q;i} = \left.\frac{\partial^2 \widehat{\gamma}_i}{\partial s \partial Q}\right|_{s_0;Q_0} = [4\widehat{\theta_s}\widehat{\theta_Q}(s_0 - s_i)(Q_0 - Q_i)^2]\exp[-\widehat{\theta_s}(s_0 - s_i)^2 - \widehat{\theta_Q}(Q_0 - Q_i)^2]$$

$$\kappa_{Q;i} = \left.\frac{\partial^2 \widehat{\gamma}_i}{\partial Q^2}\right|_{s_0;Q_0} = [-2\widehat{\theta_Q} + 4\widehat{\theta_Q}^2(Q_0 - Q_i)^2]\exp[-\widehat{\theta_s}(s_0 - s_i)^2 - \widehat{\theta_Q}(Q_0 - Q_i)^2]$$

$$h_s = \boldsymbol{\kappa}_s^T \widehat{\boldsymbol{\Gamma}}^{-1}(\mathbf{w} - \widehat{\mu}\mathbf{1}) \quad h_{s;Q} = \boldsymbol{\kappa}_{s;Q}^T \widehat{\boldsymbol{\Gamma}}^{-1}(\mathbf{w} - \widehat{\mu}\mathbf{1}) \quad h_Q = \boldsymbol{\kappa}_Q^T \widehat{\boldsymbol{\Gamma}}^{-1}(\mathbf{w} - \widehat{\mu}\mathbf{1}) \tag{6}$$

so the Hessian is the symmetric 2×2 matrix with the off-diagonal element $\partial^2 \widehat{\widehat{C}}/\partial s \partial Q$ at (s_0, Q_0).

Unfortunately, we find that only six of the n = 20 old points give PSD Hessians; namely, those point that satisfy the subarea $0 \leq s_0 \leq 55$ and $0 \leq Q_0 \leq 49$; see the last column of Table 1. We offer two explanations:

1. $E(C) = f(s, Q)$ (the true I/O function in Kroese, Taimre, and Botev (2011)'s simulation) is not convex. The conditions for a convex function in (s, S) systems are derived by Sahin (1982); e.g., lead times are constant and the demand distribution must belong to certain families (e.g. exponential). However, when in Kroese, Taimre, and Botev (2011)'s simulation we make the lead times constant and the demand distribution exponential, we still find points that are not PSD (we do not display these results).
2. Even if $E(C)$ satisfies the convexity conditions, we estimate its convexity through a Kriging metamodel that is not convex (but wiggles). Wiggling Kriging is known to result if the Kriging ignores the randomness (internal noise, nugget) of the simulation output; see Figure 2 in Yin, Ng, and Ng (2011). "Stochastic" Kriging accounts for this randomness, so Kriging is no longer an exact interpolator—which may eliminate wiggling. We use DACE, as we do for monotonic Kriging. Moreover, Kriging in deterministic simulation is known to be a bad extrapolator; the points in Table 1 that do not give PSD Hessians are near the border of the experimental area.

Next we run a new experiment that is limited to the *subarea* $0 \leq s_0 \leq 55$ and $0 \leq Q_0 \leq 49$. Fitting a Kriging metamodel gives the plot in Figure 2. Unfortunately, we again find that only ten (was six) of the twenty "old" points in this subarea give PSD Hessians. We also estimate the Hessians at 55×49 new points on a grid. Altogether we find PSD Hessians for the sub-subarea $0 \leq s_0 \leq 36$ and $0 \leq Q_0 \leq 21$.

Given these problematic results for the inventory simulation, we decide to examine our bootstrapped convex Kriging through an *artificial (Monte Carlo) example*. This example is inspired by this simulation; i.e., to the I/O data for the subarea $0 \leq s_0 \leq 55$ and $0 \leq Q_0 \leq 49$ we fit a *second-order polynomial* in x_1 and x_2 instead of s and Q (= S - s). For this fitting we use ordinary least squares (OLS). We treat these OLS estimates as the true coefficients, so the artificial example becomes:

$$E[y(x_1, x_2)] = 332.794 - 7.427x_1 - 4.922x_2 + 0.127x_1^2 + 0.093x_2^2 + 0.130x_1x_2. \tag{7}$$

It is easy to check that this function has a PSD Hessian so it is convex. Its optimal input combination is $\mathbf{x}_{opt} = (x_{1;opt}, x_{2;opt}) = (24.5, 9.2)$, which gives the optimal output y_{opt} = 218.7.

Next we fit a Kriging metamodel to the same twenty input combinations inside the subarea $0 \leq s_0 \leq 55$ and $0 \leq Q_0 \leq 49$ selected through LHS (not displayed); i.e., we use $x_{1;i} = s_i$ and $x_{2;i} = S_i$ (i = 1, ..., 20) but we replace $\overline{C_i}$ by $E[y(x_{1;i}, x_{2;i})] = E(y_i)$ following from (7). This Kriging metamodel turns out to give PSD Hessians at sixteen of the twenty old points.

Subsequently, we make this artificial example more realistic by making it give *random* outputs; i.e., to (7) we add Gaussian noise $\epsilon_{i;r}$ with zero mean and standard deviation $\sqrt{5000}\widehat{\sigma}_i$ with $\widehat{\sigma}_i = s(\overline{C_i})$ computed

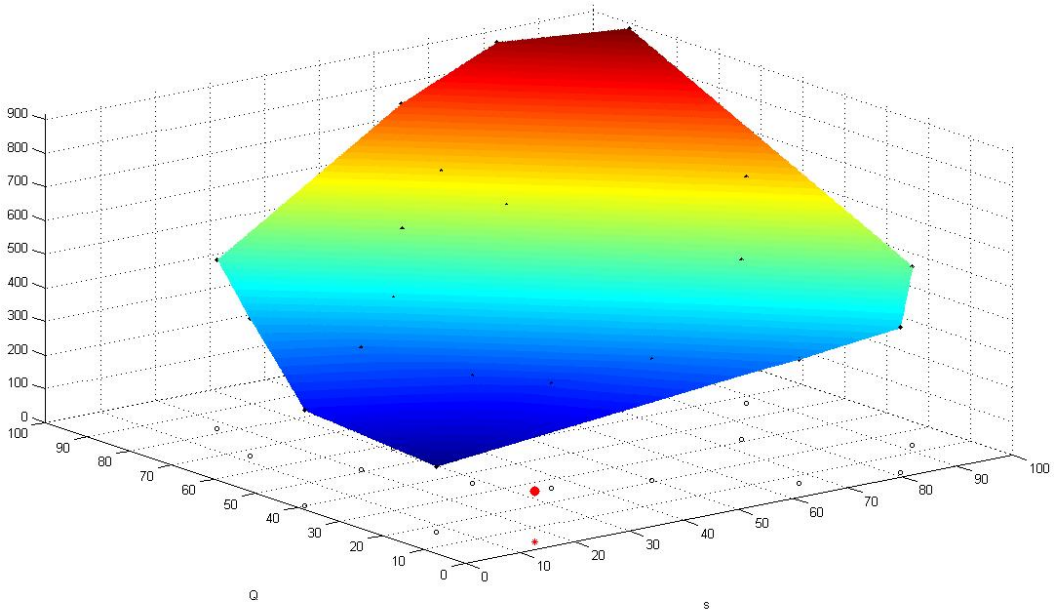

Figure 2: Kriging predictions for (s, S) simulation in subarea $0 \leq s_0 \leq 55$ and $0 \leq Q_0 \leq 49$.

from 5000 replicates $C_{i;r}$ (r = 1, ..., m) for each of the 20 points inside the subarea $0 \leq s_0 \leq 55$ and $0 \leq Q_0 \leq 49$ ($\widehat{\sigma}_i$ not displayed):

$$y_{i;r} = E(y_i) + \epsilon_{i;r} \ (i = 1, \ldots, n) \ (r = 1, \ldots, m). \tag{8}$$

To make the example more representative of expensive simulations, we select the number of replications much smaller than 5000; namely, m = 10 (so the variance of the average output $\overline{y_i}$ increases).

We fit a Kriging metamodel to the n averages $\overline{y_i} = \sum_{r=1}^{m} y_{i;r}/m$, using DACE. We find that this Kriging gives PSD Hessians at only eight of the twenty old points.

For a wiggling original Kriging metamodel, we *bootstrap*. So, for input combination $(x_{1;i}, x_{2;i})$ we resample—with replacement—the m original outputs $y_{i;r}$ (see (8)) to obtain the bootstrapped simulation outputs $y^*_{i;r}$ and their average $\overline{y^*_i} = \sum_{r=1}^{m} y^*_{i;r}/m$. We do so for each of the n combinations, which gives the vector of bootstrap averages $\overline{\mathbf{y}^*} = (\overline{y^*_1}, \ldots, \overline{y^*_n})^T$.

Next we fit a Kriging model to $(\mathbf{X}, \overline{\mathbf{y}^*})$ where $\mathbf{X}$ is the 20×2 matrix of input combinations $(x_{1;i}, x_{2;i})$. This bootstrapped Kriging model gives predictions y^*, which differ from the original predictions $\widehat{\widehat{y}}$, because (with probability 1) $\overline{\mathbf{y}} \neq \overline{\mathbf{y}^*}$ and $\widehat{\theta} \neq \widehat{\theta}^*$. We *accept* only those bootstrapped Kriging metamodels that have at least as many old points with PSD Hessians as the original Kriging metamodel has; i.e., at least eight PSD points.

After some experimentation with the bootstrap sample size B, we report results for B = 1000. This gives B_a = 418 accepted bootstrapped Kriging metamodel with at least 8 out of 20 Hessians being PSD (the classic Kriging metamodel had 8 PSD old points). (The maximum number of PSD Hessians in the accepted metamodels is 16; this maximum occurs in bootstrap b_a = 97.)

We expect that the accepted Kriging metamodels improve *simulation optimization*. There are many simulation–optimization methods, but we apply a simple *grid search*; i.e., in the area of interest ($0 \leq x_1 \leq 55$ and $0 \leq x_2 \leq 49$) we compute the Kriging predictor at (say) the 56×50 grid of integers, and select the combination that gives the minimum predicted output y^*. So, the B_a = 418 accepted Kriging metamodels give the estimated optimum outputs $y^*_{b;opt}$ with b = 1, ..., 418. To get CIs, we sort these estimates; the

resulting order statistics $y^*_{(b);opt}$ give the 90% CI $[y^*_{(21);opt}, y^*_{(397);opt}] = [191.65, 304.88]$. They also show one outlier; namely, $y^*_{(1);opt}$ = -283.00. The median is $y^*_{(209);opt}$ = 287.83. Furthermore, $\widehat{\widehat{y_{opt}}}$ = 303.012 (the result of the grid search applied to the original Kriging metamodel $\widehat{\widehat{y}}$) and y_{opt} = 218 (true optimum following from the second-order polynomial (7)).

The B_a = 418 metamodels also give the estimated optimum input combinations $\mathbf{x}^*_{b;opt} = (x^*_{b;1;opt}, x^*_{b;2;opt})^T$ with b = 1, ..., 418. Sorting these estimates for the optimal input x_1 gives the order statistics $x^*_{(b);1;opt}$, which give the 90% CI $[x^*_{(21);1;opt}, x^*_{(397);1;opt}] = [21, 42]$. The median is $x^*_{(209);1;opt}$ = 39. Furthermore, $\widehat{\widehat{x_{1;opt}}}$ = 38 (for original Kriging metamodel $\widehat{\widehat{y}}$) and $x_{1;opt}$ = 24.5 (true optimum input of second-order polynomial (7)). Likewise, for x_2 we obtain the 90% CI $[x^*_{(21);2;opt}, x^*_{(397);2;opt}] = [4, 25]$, median $x^*_{(209);2;opt}$ = 18, $\widehat{\widehat{x_{2;opt}}}$ = 21, and $x_{2;opt}$ = 9.2.

In this artificial example we know the *true* I/O function, so we can verify the preceding results; i.e., into (7) we substitute $\widehat{\widehat{\mathbf{x}_{opt}}} = (38, 21)^T$ (optimal combination estimated through the original Kriging model), $\mathbf{x}^*_{b_a;opt}$ (optimal combination estimated through accepted bootstrapped Kriging model b_a). This gives $y(\widehat{\widehat{\mathbf{x}_{opt}}}) = y(38, 21)$ = 274.918 and $y(\mathbf{x}^*_{b_a;opt})$ with b_a = 1, ..., 418; these $y(\mathbf{x}^*_{b_a;opt})$ range between 218.947 and 310.888 (remember y_{opt} = 218). (The bootstrapped metamodel with the maximum number of PSD Hessians gives $y(\mathbf{x}^*_{97;opt})$ = 278.339.) We point out that $\mathbf{x}_{opt}$ (true optimal combination) and $\widehat{\widehat{\mathbf{x}_{opt}}}$ (classic estimator) lie within the rectangle defined by the CIs for the two optimal inputs.

We conclude that in this artificial example our bootstrapping helps find better solutions than classic Kriging suggests. Specifically, the CIs for the optimal inputs suggest that in the next stage we should simulate and search in the subarea $21 \leq x_1 \leq 42$ and $4 \leq x_2 \leq 25$ (the experimental area was $0 \leq x_1 \leq 55$ and $0 \leq x_2 \leq 49$)**.**

5 CONCLUSIONS AND FUTURE RESEARCH

In practice, simulation may be computationally expensive, so we simulate only a few input combinations and replicate these combinations only a few times. Classic Kriging may then give metamodels that contradict our *prior qualitative* (structural) knowledge of the characteristics (e.g., convexity or monotonicity) of the I/O function that is implicitly defined by underlying simulation model. Users may then reject the metamodel and the simulation model, and we (as analysts) may find that the metamodel does not provide good sensitivity analysis or does not accurately estimate the true optimum I/O of the simulated system.

Our *monotonic* distribution-free bootstrapped Kriging for an M/M/1 simulation turns out to give better coverage without longer CI. Unfortunately, this coverage may still be lower than desired, because the small number of simulation observations may give too little information to estimate an adequate metamodel—be that metamodel a classic Kriging or a monotonic bootstrapped Kriging metamodel. In such situations we would advise spending more computer time to obtain reliable results, but while awaiting these results we can bootstrap the too small sample to obtain a monotonic bootstrapped Kriging metamodel that is better than the classic Kriging metamodel.

An additional advantage of our bootstrapped Kriging is that the CI does not include negative values if negative values are impossible (as in the simulation of waiting times). Technically, bootstrapped Kriging does not give an exact interpolator, which is attractive because the average simulation outputs still show sampling variation. Our Kriging also allows variance heterogeneity of the simulation outputs.

We also try to derive *convex distribution-free bootstrapped Kriging*. A twice differentiable convex function implies PSD Hessians at all input combinations. Unfortunately, a given simulation model defines its I/O function only implicitly. Therefore we fit a Kriging metamodel to the simulation I/O data; this Kriging metamodel implies estimated Hessians at old and new points. We verified that fitted Kriging metamodels may show Hessians that are not PSD at several old points, even in our example of a second-order polynomial without noise and with coefficients such that this polynomial has PSD Hessians. In random simulation

we obtain replicates for all old points (to improve the accuracy of the simulated output), so we can apply distribution-free bootstrapping to these replicates. To these bootstrapped outputs we can apply Kriging. We accept only those bootstrapped Kriging metamodels that have at least as many old points with PSD Hessians as the original Kriging metamodel.

We illustrate our bootstrapped Kriging through two types of examples: (i) a (s, S) inventory simulation, and (ii) an artificial Monte Carlo experiment with a convex second-order polynomial augmented with Gaussian noise. Example (i) gives a Kriging metamodel that was not convex; i.e., some old points gives non-PSD Hessians. Example (ii) demonstrated that accepting those bootstrapped Kriging models with at least the same number of PSD Hessians gives CIs that do cover the true optimal input combination; classic Kriging does not give a CI for the optimal input (it does give a CI for the output of a given input combination). These CIs may limit the area in which we search for the optimum, in a next stage (which is not the topic of this paper).

Future research may try to solve the following problems:

- Extension to "stochastic Kriging", formalized by Ankenman, Nelson, and Staum (2010), Chen, Ankenman, and Nelson (2010), and Yin, Ng, and Ng (2011). This stochastic Kriging covers a nugget effect with homogeneous (constant) variances, a modified nugget effect with heterogeneous variances, and nugget effects that in case of CRN are correlated across input combinations. For software we also refer to Dancik and Dorman (2008), Roustan, Ginsbourger, and Deville (2012), and Rasmussen and Nickisch (2012).
- Replacement of Ordinary Kriging by Universal Kriging, which replaces the constant term by a first-order and a second-order polynomial respectively (Universal Kriging turned out not to remove the wiggling in the M/M/1 example, and to give excellent results for the second-order polynomial example).
- Replacement of the simple grid search by one or more popular simulation-optimization methods (e.g., response surface methodology or RSM, efficient global optimization or EGO, a genetic algorithm).
- Extension of our approach to $k > 2$ inputs, including practical applications (e.g., supply chains).
- Preservation of structural knowledge about other characteristics of the I/O function (besides monotonicity and convexity); e.g., Kriging predictions may be required to be nonnegative.
- Bootstrapping other metamodeling methods (besides Kriging); e.g., isotonic regression.

REFERENCES

Ankenman, B. E., B. L. Nelson, and J. Staum. 2010. "Stochastic kriging for simulation metamodeling". *Operations Research* 58:371–382.

Boyd, S., and L. Vandenberghe. 2004. *Convex Optimization*. Berichte über verteilte messysteme. Cambridge University Press.

Chen, X., B. Ankenman, and B. Nelson. 2010, December. "Common random numbers and stochastic kriging". In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, 947 –956. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Dancik, G. M., and K. S. Dorman. 2008. "mlegp: statistical analysis for computer models of biological systems using R". *Bioinformatics* 24 (17): 1966–1967.

Efron, B., and R. Tibshirani. 1993. *An Introduction to the Bootstrap*. Monographs on Statistics and Applied Probability. Chapman & Hall.

Kleijnen, J. P. C. 2008. *Design and analysis of simulation experiments*. Springer-Verlag. (Chinese translation published by Publishing House of Electronics Industry, Beijing, 2010).

Kleijnen, J. P. C., and W. C. M. van Beers. 2011. "Monotonicity-preserving bootstrapped Kriging metamodels for expensive simulations". *Journal of the Operational Research Society*. Accepted.

Kroese, D. P., T. Taimre, and Z. I. Botev. 2011. *Handbook of Monte Carlo Methods (Wiley Series in Probability and Statistics)*. 1 ed. Wiley.

Kroese, D. P. 2012. "Handbook of Monte Carlo Methods Web Page". Accessed May. 24, 2012. http://www.maths.uq.edu.au/~kroese/montecarlohandbook/.

Loeppky, J. L., J. Sacks, and W. J. Welch. 2009. "Choosing the Sample Size of a Computer Experiment: A Practical Guide". *Technometrics* 51:366–376.

Lophaven, S., H. Nielsen, and J. Sondergaard. 2002. *DACE: a MATLAB Kriging toolbox, version 2.0*. Lyngby, Denmark: IMM Technical University of Denmark.

Nesterov, Y. 2003. *Introductory Lectures on Convex Optimization: A Basic Course (Applied Optimization)*. 1 ed. Springer.

Rasmussen, C. E. and H. Nickisch 2012. "Documentation for GPML Matlab Code version 3.1". Accessed July. 3, 2012. http://www.gaussianprocess.org/gpml/code/matlab/doc/index.html.

Roustan, O., D. Ginsbourger, and Y. Deville. 2012. "DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization". *Journal of Statistical Software*. Accepted.

Sahin, I. 1982. "On the Objective Function Behavior in (s, S) Inventory Models". *Operations Research* 30 (4): pp. 709–724.

Santner, T. J., B. J. Williams, and W. I. Notz. 2003. *The Design and Analysis of Computer Experiments*. New York: Springer-Verlag.

Yin, J., S. Ng, and K. Ng. 2011. "Kriging meta-model with modified nugget effect: an extension to heteroscedastic variance case". *Computers and Industrial Engineering* 61 (3): 760–777.

ACKNOWLEDGMENTS

We thank two anonymous referees for their very knowledgeable comments that lead to technical and stylistic corrections.

AUTHOR BIOGRAPHIES

JACK P.C. KLEIJNEN is Professor of "Simulation and Information Systems" at Tilburg University, where he is a member of both the Department of Information Management and the Operations Research Group of the Center for Economic Research (CentER) in the Tilburg School of Economics and Management (TiSEM). His research concerns the statistical design and analysis of experiments with simulation models, in many scientific disciplines (e.g., management, economics, and engineering). He was a consultant for several organizations in the USA and Europe. He serves on many international editorial boards and scientific committees. He spent several years in the USA, at universities and private companies. He received a number of national and international awards; e.g., in 2008 he received a knighthood and in 2005 an LPAA. His email address is Kleijnen@tilburguniversity.edu and his web page is http://www.tilburguniversity.edu/webwijs/show/?uid=kleijnen.

EHSAN MEHDAD is a Ph.D. student at Tilburg University. He received his Master degree in Operations Research from Tilburg University. His research interests are in discrete-event simulation and Kriging. His email address is E.Mehdad@uvt.nl.

WIM C.M. VAN BEERS is an Assistant Professor in the Department of Quantitative Economics of the University of Amsterdam. His primary research area is Kriging in random simulation. His email address is W.C.M.vanBeers@uva.nl and his web page is http://home.medewerker.uva.nl/w.c.m.vanbeers.