Customized Sequential Designs for Random Simulation Experiments

van Beers, W.C.M.; Kleijnen, J.P.C.

Publication date:
2004

No. 2004–63

CUSTOMIZED SEQUENTIAL DESIGNS FOR RANDOM SIMULATION EXPERIMENTS: KRIGING METAMODELING AND BOOTSTRAPPING

By W.C.M. van Beers, J.P.C. Kleijnen

July 2004

ISSN 0924-7815

Seq__Design_Boot_07.doc; written: 25 June 2004; printed: 7/2/2004 11:16 AM

Customized Sequential Designs for Random Simulation Experiments: Kriging Metamodeling and Bootstrapping

Wim C.M. van Beers [1] and Jack P.C. Kleijnen [2]

[1] Department of Information Systems and Management
Tilburg University (UvT), Postbox 90153, 5000 LE Tilburg, The Netherlands
Phone: +31-13-4668202; Fax: +31-13-4663069; E-mail: wvbeers@uvt.nl

[2] Department of Information Systems and Management/
Center for Economic Research (CentER)
Tilburg University (UvT), Postbox 90153, 5000 LE Tilburg, The Netherlands
Phone: +31-13-4668202; Fax: +31-13-4663069; E-mail: kleijnen@uvt.nl
http://center.uvt.nl/staff/kleijnen/

Customized Sequential Designs for Random Simulation Experiments: Kriging Metamodeling and Bootstrapping

Wim C.M. van Beers and Jack P.C. Kleijnen

Subject classification

Simulation: design of experiments, statistical analysis, Kriging, bootstrapping, regression

Abstract

This paper proposes a novel method to select an experimental design for interpolation in random simulation. (Though the paper focuses on Kriging, this method may also apply to other types of metamodels such as linear regression models.) Assuming that simulation requires much computer time, it is important to select a design with a small number of observations (or simulation runs). The proposed method is therefore sequential. Its novelty is that it accounts for the specific input/output behavior (or response function) of the particular simulation at hand; i.e., the method is customized or application-driven. A tool for this customization is bootstrapping, which enables the estimation of the variances of predictions for inputs not yet simulated. The new method is tested through the classic M/M/1 queueing simulation. For this simulation the novel design indeed gives better results than a Latin Hypercube Sampling (LHS) with a prefixed sample of the same size.

1. Introduction

In this paper, we focus on *expensive simulations*; that is, we assume that a single simulation run takes 'much' computer time. Consequently, 'interpolation' is needed; i.e., from the simulated input/output (I/O) data, the outputs are predicted for input combinations not yet simulated. We devise a method that is meant to minimize the number of simulation runs for such interpolation. We *tailor* our design of experiments (DOE) to the actual simulation; that is, we do not derive a generic design such as a classic design (for example, a 2^{k-p} design) or a LHS design. The differences between customized designs and generic designs are explained by Kleijnen and Van Beers (2004), as follows.

A *metamodel* is a model of the I/O function (or 'response function') implied by the underlying simulation model. We denote the metamodel by $Y(\mathbf{x})$ where $\mathbf{x}$ denotes the *k*-dimensional vector of the *k* inputs (factors) so $\mathbf{x} = (x_1, \cdots, x_j, \cdots, x_k)'$. *Classic DOE* assumes a simple metamodel. For example, designs of resolution III (including certain 2^{k-p} designs) assume a first-order polynomial I/O function. Composite designs (CCD) assume a second-order polynomial. These designs are discussed for physical experiments in (for example) the well-known textbook Box, Hunter, and Hunter (1978) and the recent textbook Myers and Montgomery (2002); for simulation experiments we refer to Kleijnen (1987).

*LHS (*much applied in Kriging, described below) assumes that an adequate metamodel is more complicated than a low-order polynomial. LHS, however, does not assume a specific metamodel. Instead, LHS focuses on the design space formed by the *k*–dimensional unit cube, defined by $0 \le x_j \le 1$ ($j = 1, \ldots, k$) after standardizing (scaling) the inputs. LHS is one of the *space filling* designs: LHS samples that space according to some prior distribution for the inputs, such as independent uniform distributions on [0, 1]; see McKay, Beckman, and Conover (1979, 2000), and also Kleijnen et al. (2004), Koehler and Owen (1996), and Santner, Williams, and Notz (2003).

Unlike LHS, we explicitly account for the I/O function. Unlike classic DOE, we assume that a low-order polynomial (estimated through regression analysis) gives an inadequate approximation of the I/O function. We therefore estimate the uncertainty of predicted outputs at unobserved input combinations (these combinations are also called scenarios, design points, combinations of factor levels or simulation inputs). To estimate the uncertainty of these predictions, we use *bootstrapping*; i.e., we resample the outputs for each scenario already simulated (for bootstrapping in general see the classic textbook, Efron and Tibshirani 1993; for bootstrapping in the validation of regression metamodels in simulation see Kleijnen and Deflandre 2004).

We make our procedure *sequential* for the following two reasons.

1. Sequential statistical procedures are known to be more 'efficient'; that is, they require fewer observations than fixed-sample (one-shot) procedures; see, for example, the handbook by Ghosh and Sen (1991) and the recent article by Park et al. (2002).
2. Simulation experiments proceed sequentially (unless parallel computers are used; our procedure is well suited for parallel computers).

The literature on *deterministic* simulation shows several designs that—like ours—account for the specific simulation's I/O function, and are sequential. For example, Crary (2002) discusses G-optimal and I-optimal designs, which the DOE literature defines as follows. G-optimal designs minimize the *maximum* Mean Squared Error (MSE) of the predicted output; I-optimal or Integrated MSE (IMSE) designs minimize the *average* MSE (obviously, the MSE reduces to the variance if the predictor is unbiased; see (5) and (6) below). Williams, Santner, and Notz (2000, 2002) use a Bayesian approach to derive sequential IMSE designs. Sasena, Papalambros, and Govaerts (2002) derive sequential designs for the optimisation of deterministic simulation models. Kleijnen and Van Beers (2004) derive customized sequential designs for deterministic simulations. We, however,

focus on DOE for random simulations, and we seem to be the first to apply bootstrapping for this problem.

We shall see that our designs concentrate on input combinations in sub-areas that have *more interesting* I/O behavior. In our example, we spend most of our computer simulation time on the challenging 'explosive' part of the metamodel that estimates the mean steady-state waiting time for various traffic rates of single-server queueing systems with Markovian (Poisson) arrival and service times (M/M/1 systems). (The reader may take a peek at Figure 1 discussed in section 5.) In this example, we compare our customized sequential designs with classic fixed LHS; our design gives better predictions.

We summarize our paper as follows. As a metamodel for interpolation, we use Kriging instead of linear or nonlinear regression. To estimate the parameters of this metamodel, we need a criterion for selecting a design; we do not use D-optimality or a related criterion used in classic DOE for regression metamodels, but we use a prediction error criterion, traditional in DOE for Kriging. To sequentially select candidate input combinations for actual simulation, we apply distribution-free bootstrapping. To validate our approach, we simulate the classic M/M/1 model.

The remainder of this paper is organized as follows. Section 2 summarizes the basics of Kriging. Section 3 summarizes DOE and Kriging. Section 4 explains our method, which uses bootstrapping—to estimate the variances of the Kriging predictions for candidate inputs not yet simulated—and sequentially selects as the next input to be simulated the one with the largest bootstrap variance. Section 5 demonstrates the procedure through M/M/1 simulations, which show that our method gives better results than LHS with a prefixed sample size. Section 6 present conclusions and topics for further research.

2. Kriging basics

Kriging (named after the South-African mining engineer Krige) is an interpolation method that predicts unknown values of a random function or random process; see Journel and Huijbregts (1978) and Cressie (1993)'s classic Kriging textbook on spatial (geo)statistics. Whereas spatial statistics considers the two-dimensional 'location' as the known input of this process, simulation considers the k –dimensional 'scenario' as input; see Sacks et al. (1989)'s classic article on the Design and Analysis of Computer Experiments (DACE)—these computer experiments concern deterministic simulation. Random (stochastic) simulation—including Discrete Event Dynamic Systems (DEDS) simulations—is the topic of this paper.

More precisely, a Kriging prediction is a weighted linear combination of all output values already observed. These weights depend on the distances between the new input to be predicted and the old inputs already observed. Kriging assumes that *the closer the inputs are, the more positively correlated the outputs are*. Mathematical formulations follow in equations (1) through (4).

Nowadays, Kriging is frequently applied in *deterministic simulation*, which is much used in engineering; again see Sacks et al. (1989); for an update see Simpson et al. (2001). In deterministic simulation, Kriging has an important advantage over regression analysis: the predicted values at old inputs are exactly equal to the observed (simulated) outputs.

In *random simulation*, however, this advantage disappears. Now, each scenario is simulated several times—with non-overlapping pseudo-random number (PRN) streams. Van Beers and Kleijnen (2003) show that Kriging interpolates the *average* output per scenario. These averages, however, are still random, so the fact that at simulated scenarios the Kriging predictions equal the averages loses its intuitive appeal. Still, Kriging may be attractive because it may decrease the prediction *bias* (and hence the MSE) at scenarios close together. Indeed, in the examples presented by Van Beers and Kleijnen (2003) the Kriging predictions

are much better than the regression predictions (regression analysis may be useful for other goals such as screening and validation; see Kleijnen et al. 2004). Therefore we do not further discuss regression analysis in this paper.

Mathematically formulated, Kriging assumes the following metamodel:

$$Y(\mathbf{x}) = \mu(\mathbf{x}) + \delta(\mathbf{x}) \text{ with } \delta(\mathbf{x}) \sim \text{IID}(0, \sigma^2(\mathbf{x})) \tag{1}$$

where μ is the mean of the stochastic process $Y(\cdot)$, and $\delta(\mathbf{x})$ is the additive *noise*, which is assumed independently and identically distributed (IID) with mean zero and variance $\sigma^2(\mathbf{x})$. 'Ordinary' Kriging—to which we limit ourselves—further assumes a *stationary covariance process* for $Y(\mathbf{x})$ in (1); i.e., the expected values $\mu(\mathbf{x})$ are a constant μ and the covariances of $Y(\mathbf{x}+\mathbf{h})$ and $Y(\mathbf{x})$ depend only on the distance (lag) $|| \mathbf{h} || = || (\mathbf{x}+\mathbf{h}) - (\mathbf{x}) ||$.

The Kriging *predictor* for the unobserved (non-simulated) input (say) $\mathbf{x}_0$—denoted by $\hat{Y}(\mathbf{x}_0)$—is a weighted linear combination of all the *n* observed outputs:

$$\hat{Y}(\mathbf{x}_0) = \sum_{i=1}^{n} \lambda_i \cdot Y(\mathbf{x}_i) = \boldsymbol{\lambda}' \cdot \mathbf{Y} \tag{2}$$

with $\sum_{i=1}^{n} \lambda_i = 1$, $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)'$ and $\mathbf{Y} = (y_1, \ldots, y_n)'$. To choose these weights, Kriging derives the Best Linear Unbiased Predictor (BLUP), which minimizes the MSE of the predictor:

$$\min_{\lambda} \left\{ \text{MSE}\left(\hat{Y}(\mathbf{x}_0)\right) \right\} = \min_{\lambda} \left\{ E\left(Y(\mathbf{x}_0) - \hat{Y}(\mathbf{x}_0)\right)^2 \right\}. \tag{3}$$

Obviously, this solution depends on the output's covariances. It can be proven that the optimal weights in (2) resulting from (3) are

$$\boldsymbol{\lambda}' = \left(\boldsymbol{\gamma} + \mathbf{1}\frac{1 - \mathbf{1}'\boldsymbol{\Gamma}^{-1}\boldsymbol{\gamma}}{\mathbf{1}'\boldsymbol{\Gamma}^{-1}\mathbf{1}} \right)' \boldsymbol{\Gamma}^{-1} \tag{4}$$

with the following symbols:

$\boldsymbol{\gamma}$ is the vector of covariances between the outputs at the input to be predicted and at the already observed inputs, so $\boldsymbol{\gamma} = (\gamma(\mathbf{x}_0 - \mathbf{x}_1), \ldots, \gamma(\mathbf{x}_0 - \mathbf{x}_n))'$;

$\mathbf{1} = (1, \ldots, 1)'$ is the vector of ones;

$\boldsymbol{\Gamma}$ is the $n \times n$ matrix whose element (i, j) is the (co)variance at the already observed inputs $\gamma(\mathbf{x}_i - \mathbf{x}_j)$ with $i, j = 1, \ldots, n$.

We point out that the weights in (4) vary with $\mathbf{x}_0$ (input to be predicted), whereas regression analysis uses the same estimated metamodel for all inputs $\mathbf{x}$.

We further observe that the literature on (deterministic) simulation speaks of covariances and corresponding correlations, whereas the geostatistics literature speaks of the *variogram*, defined as $2\gamma(\mathbf{h}) = var(Y(\mathbf{x} + \mathbf{h}) - Y(\mathbf{x}))$. Since we shall use the Matlab Kriging toolbox DACE—made available free of charge by Lophaven, Nielsen, and Søndergaard (2002)—we avoid the term variogram.

We emphasize that in practice the covariances $\boldsymbol{\gamma}$ and $\boldsymbol{\Gamma}$ in (4) must be *estimated*. Consequently, the weights in (4) become random variables (say) $\hat{\boldsymbol{\lambda}}$. These weights make the Kriging predictor resulting from (2) *non-linear*. This characteristic is often neglected in the Kriging literature. In general, non-linear variables are hard to analyze—a simple computer-intensive solution is bootstrapping; see Efron and Tibshirani (1993).

Ignoring the randomness of the estimated optimal weights $\hat{\lambda}$ tends to *underestimate* the true variance of the Kriging predictor. This follows from the general formula for the conditional variance $\text{var}(Y \mid X) = (1 - \rho^2) \cdot \text{var}(Y)$; see, for example, Kreyszig (1970, p. 343). To tackle this problem, Cressie (1993, p. 146) proposes *cross-validation*. Cross-validation is also used by Kleijnen and Van Beers (2004) for deterministic simulation. For deterministic simulation Den Hertog, Kleijnen, and Siem (2004) apply parametric bootstrapping—assuming normally distributed prediction errors—and find that ignoring the randomness of the Kriging weights leads to serious errors. Because random simulation may have non-normal outputs (for example, queueing simulations have distributions with heavy right-hand tails), we use distribution-free bootstrapping—as we shall explain.

3. DOE and Kriging

By definition, an experimental *design* is a set of *n* combinations of *k* factor values. These combinations are usually bounded by 'box' constraints: $a_j \le x_j \le b_j$ where $a_j, b_j \in R$ with $j = 1, \ldots, k$. The set of all feasible combinations is called the *experimental region* (say) *H*. We suppose that *H* is a *k*-dimensional unit cube, after rescaling the original rectangular area (see the Introduction, Section 1).

Our goal is to find the 'best' design for Kriging predictions within *H*; the Kriging literature proposed several criteria (see Sacks et al. 1989, p. 414). Most of these criteria are based on the predictor's MSE. Most progress has been made for the IMSE (see Bates et al. 1996):

$$IMSE = \int_H \text{MSE}\left(\hat{Y}(\mathbf{x})\right) \phi(\mathbf{x}) d\mathbf{x} \tag{5}$$

where MSE follows from (3), and $\phi(\mathbf{x})$ is a given weight function—usually assumed to be uniform.

To evaluate a design, Sacks et al. (1989, p. 416) compare the predictions with the known output values of a *test set* consisting of (say) *m* inputs. The IMSE in (5) can then be estimated by the Empirical IMSE (EIMSE):

$$EIMSE = \frac{1}{m}\sum_{i=1}^{m}\left(\hat{y}_i(\mathbf{x}) - y_i(\mathbf{x})\right)^2 . \qquad (6)$$

Besides this EIMSE, we will also study the *maximum* MSE; that is, we also consider risk-averse users (see Van Groenigen, 2000). So IMSE—defined in (5)—is replaced by

$$MaxMSE = \max_{\mathbf{x}\in H}\left\{MSE\left(\hat{Y}(\mathbf{x})\right)\right\} \qquad (7)$$

and EIMSE in (6) by

$$EMaxIMSE = \max_{i\in\{1,\ldots,m\}}\left\{\left(\hat{y}_i(\mathbf{x}) - y_i(\mathbf{x})\right)^2\right\}. \qquad (8)$$

4. Sequential DOE

To decide on the final design, we devise the following sequential procedure with eight steps.

Step 1. We start with a small *pilot design* with (say) n_0 input combinations; for example, n_0 = 5. We select the specific n_0 values such that they are equally spread over the

experimental region. There are various 'space filling' designs; for example, LHS designs. However, in our example—namely an M/M/1 queueing system—we use *maximin* designs; see Koehler and Owen (1996, p. 288). So we select the traffic rates $x_i \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ ($i = 1, \ldots, 5$).

Step 2: For each input value x_i, we initially generate (say) m_0 IID *replicates*—because bootstrapping requires IID observations; see Efron and Tibshirani (1993). To obtain IID observations in our M/M/1 simulation example, we apply *renewal* (regenerative) analysis (see, for example, Kleijnen and Van Groenendaal 1992, and Law and Kelton 2000). As 'the' renewal state, we choose the idle (empty) state. We therefore start the simulation run in the empty state—for each traffic rate x_i. Next we observe m_0 cycles—each with (random) *cycle lengths* (say) L_i (the higher x_i, the higher L_i tends to be). Besides the m_0 cycle lengths $L_{i;j}$ per traffic rate x_i, we observe the sum of the waiting times over that cycle:

$$sw_{i;j} = \sum_{t=1}^{L_{i;j}} w_{i;j;t} \quad (i = 1, \ldots, n_0;\ j = 1, \ldots, m_0). \tag{9}$$

To reduce the variance when comparing the (random) outputs for different inputs (i.e., to improve the signal/noise ratio), we use *common random numbers* (CRN). This is a popular variance reduction technique (VRT). It is well known that to reduce the variance substantially, the PRN (say) r_t may be manipulated as follows: successive random numbers are used alternatively to simulate the arrival time a and the service time s; in other words, $a_t = -\ln r_{2t-1} E(a)$ and $s_t = -\ln r_{2t} E(s)$ ($t = 1, 2, \ldots$). In the M/M/1 simulations the correlation between the average waiting times for two neighboring traffic rates turns out to be very high, namely 0.99.

To generate the PRN, we use the Matlab command 'rand'. To initialize the PRN, we set the Matlab generator (rather arbitrarily) to its initial state s_0; for example, s_0 = 10. The Matlab web site states: 'The uniform random number generator in MATLAB 5 (and above) uses a lagged Fibonacci generator, with a cache of 32 floating point numbers, combined with a shift register random integer generator. The integer generator uses shifts and exclusive OR's.' ; see (http://www.mathworks.com/support/solutions/data/8542.shtml) and also Moler (1995).

For further details on CRN, VRT, and PRN we refer to Law and Kelton (2000).

Step 3. Based on these m_0 bivariate IID outputs $(L_{i;\,j}, sw_{i;\,j})$ (j = 1, …, m_0) per input value x_i, we estimate the mean waiting times through

$$\bar{y}_i(m_0) = \frac{\sum_{j=1}^{m_0} sw_{i;\,j}}{\sum_{j=1}^{m_0} L_{i;\,j}} \quad . \tag{10}$$

This *ratio estimator* is asymptotically unbiased; for references see again Kleijnen and Van Groenendaal (1992) and Law and Kelton (2000). We do not try to improve the small-sample performance of this estimator (for example, through jackknifing—which is closely related to bootstrapping), because this estimator suffices for our Kriging metamodel.

To estimate the *precision* of the estimate defined in (10), we use the following (1- α) confidence interval per input value x_i:

$$P\left\{\bar{y}_i(m_0) - t_{m_0-1;\,1-\alpha/2} \cdot \frac{\hat{\sigma}_i / \sqrt{m_0}}{\bar{L}_i} \le E(w_i) \le \bar{y}_i(m_0) + t_{m_0-1;\,1-\alpha/2} \cdot \frac{\hat{\sigma}_i / \sqrt{m_0}}{\bar{L}_i}\right\} = 1-\alpha \tag{11}$$

where $\hat{\sigma}_i^2 = \hat{\text{var}}(sw_i) + \bar{y}_i^2 \cdot \hat{\text{var}}(L_i) - 2\bar{y}_i \cdot \hat{\text{cov}}(sw_i, L_i)$ and $\bar{L}_i = \sum_{j=1}^{m_0} L_{i;j} / m_0$; again see Kleijnen and Van Groenendaal (1992). Note that this interval does not have a *joint* (or experimentwise) probability (1- α) over all simulated input values.

Next, we add replicates one at a time—*sequential sampling*—until the desired half-width of the interval in (11) has reduced to a prefixed relative error (say) δ ; for example, δ = 0.15 (again see Kleijnen and Van Groenendaal 1992 and Law and Kelton 2000). We denote the final number of replicates per input x_i by m_i . This gives the average output $\bar{y}_i(m_i)$ per input x_i based on m_i replicates; see (10) with m_0 replaced by m_i .

Step 4. Based on these n_0 average outputs $\bar{y}_i(m_i)$ for the n_0 inputs x_i, we compute the *Kriging predictors* for the expected outputs of a new set of (say) n^c *candidate* input values x_g^c (g = 1, …, n^c). We again select these candidates in a space-filling way; in our example we choose the candidate inputs halfway between two old neighboring inputs: $x_g^c = (x_g + x_{g+1})/2$ (with $g = 1, \ldots, n_0 - 1$, so we avoid extrapolation).

By definition, the Kriging predictor is a weighted linear combination of all outputs already observed; see (2). So now Kriging weights the n_0 values already observed in steps 1 through 3:

$$\hat{y}(\mathbf{x}_g^c) = \sum_{i=1}^{n_0} \lambda_i \cdot y(\mathbf{x}_i) \qquad (12)$$

with $\sum_{i=1}^{n_0} \lambda_i$ = 1. To estimate the weights λ_i in (12), Kriging uses the old data set $(x_i, \bar{y}_i(m_i))$ (i = 1, …, n_0) . To estimate the variance of this non-linear predictor, we use bootstrapping—as follows.

Step 5. Per input x_i, we *bootstrap* the m_i bivariate IID outputs ($L_{i;\,j}, sw_{i;\,j}$); i.e., we resample—with replacement—the outputs resulting from steps 1 through 3. We denote these bootstrap observations by the superscript * (as is traditional in the bootstrap literature):

$$\{(sw^*_{i;\,1}, L^*_{i;\,1}), \ldots, (sw^*_{i;\,m_i}, L^*_{i;\,m_i})\}. \tag{13}$$

Using these bootstrapped observations and (10), we compute the bootstrap averages:

$$\bar{y}^*_i(m_i) = \frac{\sum_{j=1}^{m_i} sw^*_{i;\,j}}{\sum_{j=1}^{m_i} L^*_{i;\,j}}. \tag{14}$$

Using the I/O data $(x_i, \bar{y}^*_i(m_i))$ $(i = 1, \ldots, n_0)$ and (12), we compute the bootstrapped Kriging predictor:

$$\hat{y}^*(\mathbf{x}^c_g) = \sum_{i=1}^{n_0} \lambda^*_i \cdot \bar{y}^*(\mathbf{x}_i) \tag{15}$$

We again estimate the bootstrap weights λ^*_i in (15) by means of the Matlab Toolbox DACE; see Section 2.

We note that DACE can use a starting value for the numerical search that leads to the maximum likelihood estimator (MLE) of the Kriging weights λ^*_i in (15). As starting values we use the MLE for λ_i based on the original I/O data in (12).

Step 6. The resampling per input x_i in step 5 is repeated (say) *B* times (this *B* is called the bootstrap sample size). Hence, (13) through (15) give $\hat{y}_b^*(\mathbf{x}_g^c)$ (b = 1, …, B).

For each of the n^c candidate inputs x_g^c, we compute the bootstrap variance of the corresponding Kriging predictor:

$$\hat{\text{var}}(\hat{y}_g^{c*}) = \frac{1}{B-1}\sum_{b=1}^{B}(\hat{y}_{g;b}^{c*} - \bar{\hat{y}}_g^{c*})^2 \ . \tag{16}$$

Step 7. We determine which candidate input has the *largest* bootstrap prediction variance (16):

$$v = \arg\left(\max_{g \in \{1, \ldots, n^c\}} \left\{\hat{\text{var}}(\hat{y}_g^{c*})\right\}\right), \tag{17}$$

and we add this 'winning' input x_v^c to the old design.

Now, we run the simulation model with the input x_v^c—until we have m_0 replicates for this input. We still apply CRN (so we initialize the PRN with the seed s_0). Furthermore, we again start with the empty system as the renewal state. We continue the simulation until the confidence interval reaches the threshold δ ; see (11).

Step 8. We *repeat* the steps 4 through 7—until we have reached a stopping criterion. In other words, we bootstrap the old I/O set augmented with the candidate selected in step 7. We select a new set of candidates. For these candidates, we compute the Kriging predictors and their bootstrap variances. Alternative stopping criteria may be: (i) the computer budget has been exhausted, (ii) the project has reached its deadline, (iii) the precision of the Kriging metamodel is acceptable.

We observe that adding one point at a time—as we do in our sequential DOE—is not necessarily optimal. However, it is a simple—albeit myopic—heuristic; also see Banjevic and Switzer (2002, pp. 5-6), who refer to Ferri and Piccioni (1992).

5. M/M/1 example

An M/M/1 has as true I/O function the hyperbole

$$y = \frac{x}{1-x} \text{ with } 0 < x < 1 \tag{18}$$

where y denotes the expected mean steady-state waiting time assuming a service rate of 1, and x denotes the traffic rate .

We apply our customized sequential design procedure described in section 4, selecting the following parameters for our sequential design procedure.

Step 1: A pilot design of size $n_0 = 5$.

Step 2: $m_0 = 10$ replicates for the initial estimates of the variances; initial PRN seeds s_0 = 10 and s_0 = 12 respectively.

Step 3: Precision δ = 0.05 and 0.15 respectively for the confidence interval (11) with α = 0.01 and 0.05 respectively. For high traffic rates x (say, $x > 0.7$), long cycle lengths are more likely; in our experiments, we limit $L_{i;j}$ to 1000.

Step 6: Bootstrap sample sizes B = 50 and 100 respectively.

Step 8: Stopping criterion is reaching a total design size n = 15, 25, and 100 respectively.

Insert Figure 1

Figure 1 shows our design and a LHS design—both with 15 simulated traffic rates—and their Kriging predictions, for M/M/1 (for case 4b in Table 1; see below). LHS turns out to simulate fewer 'challenging' inputs; i.e., high traffic rates with large variances. Clearly, the LHS predictions deviate from the true output—especially for traffic rates larger than (roughly) 0.7.

To further evaluate our procedure, we use a *test set* consisting of 32 equidistant traffic rates with the following traffic rates: {0.1125, 0.1375, …, 0.8875}. Sacks et al. (1989) use similar test sets to evaluate their procedure. We compare our Kriging predictions with the 'true' outputs of the test set—using the true I/O function (18)—and calculate the prediction errors. (Both our final design and the LHS design may contain some members of the test set, but we ignore this phenomenon.)

We compare the *EIMSE* defined in (5) for our final design and for a *LHS* design with the same *n* (number of traffic rates). We use a LHS design with replicate numbers per input value that are again determined by the precision of the confidence interval (11). The replicate numbers *m* in our design and the LHS design may differ, so we calculate the *corrected EIMSE*:

$$CEIMSE = C \times \frac{1}{n_t} \sum_{i=1}^{n_t} \left(\hat{y}(x_i^t) - y(x_i^t) \right)^2 , \qquad (19)$$

where *C* is the ratio of the total number of replicates in the LHS design and in our design, n_t is the number of I/O combinations in the test set (here $n_t = 32$), and x_i^t is the i^{th} input of the test set.

Insert Table 1

This gives Table 1, which shows that in all cases our designs give 'better' predictions than LHS designs with the same size; i.e., our designs have smaller CEIMSE. However, as n increases, this advantage gets smaller; so our procedure is most attractive for expensive simulations with small sample sizes!

Risk-averse users use EMaxIMSE defined in (8). Our designs outperform LHS in 13 out of the 14 cases simulated; this exceptional, worst case is illustrated in Figure 2, which shows that the maximum error occurs at 0.8875 (the maximum traffic rate in the test set).

Insert Figure 2

Further, all numbers in this table have the same magnitude, even when the final sample size n varies. This implies that the magnitude of the individual prediction error decreases as n increases; see (19).

Table 1 also shows that the bootstrap sample size B has no systematic effect. Our explanation is that our procedure uses the bootstrap only to estimate which candidate input has the largest variance of the Kriging predictor; see (17). So we conclude that in practice the smaller size, $B = 50$, may be used. (Most bootstrap applications require the estimation of the whole distribution function, so B is much higher than 50; for example $B = 1000$.)

Finally, changes in α and δ affect the number of replicates, but this effect is incorporated in *CEIMSE* via the factor C; see (19).

Insert Figure 3

As we expected, the number of required replicates (or cycles) increases with the traffic rate. For example, if $\alpha = 0.05$ and $\delta = 0.15$, then the traffic rate $x = 0.1$ requires 489 simulation runs, whereas the traffic rate $x = 0.9$ requires the maximum number of runs, namely 1000; see Figure 3. Moreover, a cycle is likely to be longer as the traffic rate increases. For example, if $x = 0.1$ then the average cycle length $\bar{L} = 4.8$ for $m_0 = 10$ replicates; if $x = 0.9$ then $\bar{L} = 45.9$.

6. Conclusions and future research

In practice, simulation often requires much computer time per run (or replicate)—so an efficient experimental design for interpolation is desirable. In general, it is well known that sequential designs are more efficient than fixed-sample designs. Our specific sequential designs add as the input to be simulated next, the candidate input with the maximum estimated variance for its predicted output. As the predictor we use the Kriging metamodel; to estimate its variance, we use bootstrapping. We applied this procedure to the classic M/M/1 simulation, and compared its efficiency with LHS with a fixed sample size of the same magnitude. Our results clearly show that our procedure is indeed more efficient.

In future research, (asymptotic) proofs of the behavior of our procedure might be derived. Examples that are more complicated than the M//M/1 simulation may be investigated. Besides LHS, other designs with prefixed sizes may be explored; for example, min-max designs. Besides Ordinary Kriging, other metamodels may be used to analyze the I/O data. For example, the 'optimal' weights in Ordinary Kriging require that the predictor equal the average outputs at the inputs already observed; dropping this constraint implies that new Kriging software must be developed. Instead of the IMSE criterion, the maximum squared error may be chosen and the corresponding weights may be derived. Besides Kriging,

other metamodels may be used for prediction; for example, linear or nonlinear regression metamodels.

References

Banjevic, M. and P. Switzer (2002), Bayesian network designs for variance as a function of the location. Working Paper, Statistics Department, Stanford University

Bates, R.A., R.J. Buck, E. Riccomagno and H.P. Wynn (1996), Experimental design and observation for large systems. *Royal Statistical Society*. 58, no. 1, pp. 77-94

Box, G.E.P., W.G. Hunter and J.S. Hunter (1978), *Statistics for experimenters: an introduction to design, data analysis and model building*. John Wiley & Sons, Inc., New York

Crary, S.B. (2002), Design of computer experiments for metamodel generation, *Analog Integrated Circuits and Signal Processing*, 32, pp. 7-16

Cressie, N.A.C. (1993), *Statistics for spatial data*. John Wiley & Sons, Inc., New York

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

Ferri, M. and M. Piccioni (1992), Optimal selection of statistical units. *Computational Statistics & Data Analysis*, 13, pp. 47-61

Ghosh, B.K. and P.K. Sen (editors), 1991, *Handbook of sequential analysis*. Marcel Dekker, Inc., New York

Den Hertog, D., J.P.C. Kleijnen, and A.Y.D. Siem (2004) The correct Kriging variance estimated by bootstrapping. Working Paper. Department of Information Systems and Management, Tilburg University (preprint: http://center.kub.nl/staff/kleijnen/papers.html)

Journel, A.G. and C.J. Huijbregts (1978), *Mining geostatistics*, Academic Press, London

Kleijnen, J.P.C. (1987), *Statistical tools for simulation practitioners*. Marcel Dekker, Inc., New York

Kleijnen, J.P.C. and D. Deflandre (2004), Validation of regression metamodels in simulation: Bootstrap approach. Working Paper. Department of Information Systems and Management, Tilburg University (preprint: http://center.kub.nl/staff/kleijnen/papers.html)

Kleijnen, J.P.C., S.M. Sanchez, T.W. Lucas and T.M. Cioppa (2004), A user's guide to the brave new world of designing simulation experiments. Working Paper. Department of Information Systems and Management, Tilburg University (preprint: http://center.kub.nl/staff/kleijnen/papers.html)

Kleijnen, J.P.C. and W.C.M. van Beers (2004), Application-driven sequential designs for simulation experiments: Kriging metamodeling. *Journal of the Operational Research Society* (in press)

Kleijnen, J.P.C. and W. van Groenendaal (1992), *Simulation: a statistical perspective (Together with)* John Wiley, Chichester (England)

Koehler, J.R. and A.B. Owen (1996), Computer experiments. *Handbook of statistics*, by S. Ghosh and C.R. Rao, vol. 13, pp. 261-308

Kreyszig, E. (1970), *Introductory mathematical statistics: principles and methods.* John Wiley & Sons, Inc., New York

Law, A.M. and W.D. Kelton (2000), *Simulation modeling and analysis, third edition*, McGraw-Hill, Boston

Lophaven, S.N., H.B. Nielsen and J. Søndergaard (2002), A Matlab Kriging toolbox. *Technical report IMM-TR-2002-12*, Technical University of Denmark.

McKay, M.D., R.J. Beckman and W.J. Conover (1979), A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21, no. 2, pp. 239-245 (reprinted in 2000: Technometrics, 42, no. 1, pp. 55-61

Moler, C. (1995), Random thoughts. *MATLAB News & Notes*, pp. 12-13

Myers, R.H. and D.C. Montgomery (2002). Response surface methodology: process and product optimization using designed experiments; second edition. Wiley, New York

Park, S., J.W. Fowler, G.T. Mackulak, J.B. Keats, and W.M. Carlyle (2002), D-optimal sequential experiments for generating a simulation-based cycle time-throughput curve. *Operations Research*, 50, no. 6, pp. 981-990

Sacks, J., W.J. Welch, T.J. Mitchell and H.P. Wynn (1989), Design and analysis of computer experiments. *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:

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

Simpson, T.W., T.M. Mauery, J.J. Korte, and F. Mistree (2001), Kriging metamodels for global approximation in simulation-based multidisciplinary design optimization. *AIAA Journal*, 39, no. 12, 2001, pp. 2233-2241

Van Beers, W. and J.P.C. Kleijnen (2003), Kriging for interpolation in random simulation. *Journal of the Operational Research Society*, no. 54, pp. 255-262

Van Groenigen, J.W. (2000), The influence of variogram parameters on optimal sampling schemes for mapping by Kriging. *Geoderma*, no. 97, pp. 223-236

Williams, B.J., T.J. Santner, and W.I. Notz (2002), Sequential design of computer experiments for constrained optimization of integrated response functions, Working Paper. Ohio State University

Williams, B.J., T.J. Santner, and W.I. Notz (2000), Sequential design of computer experiments to minimize integrated response functions, *Statistica Sinica*, 10, 1133-1152

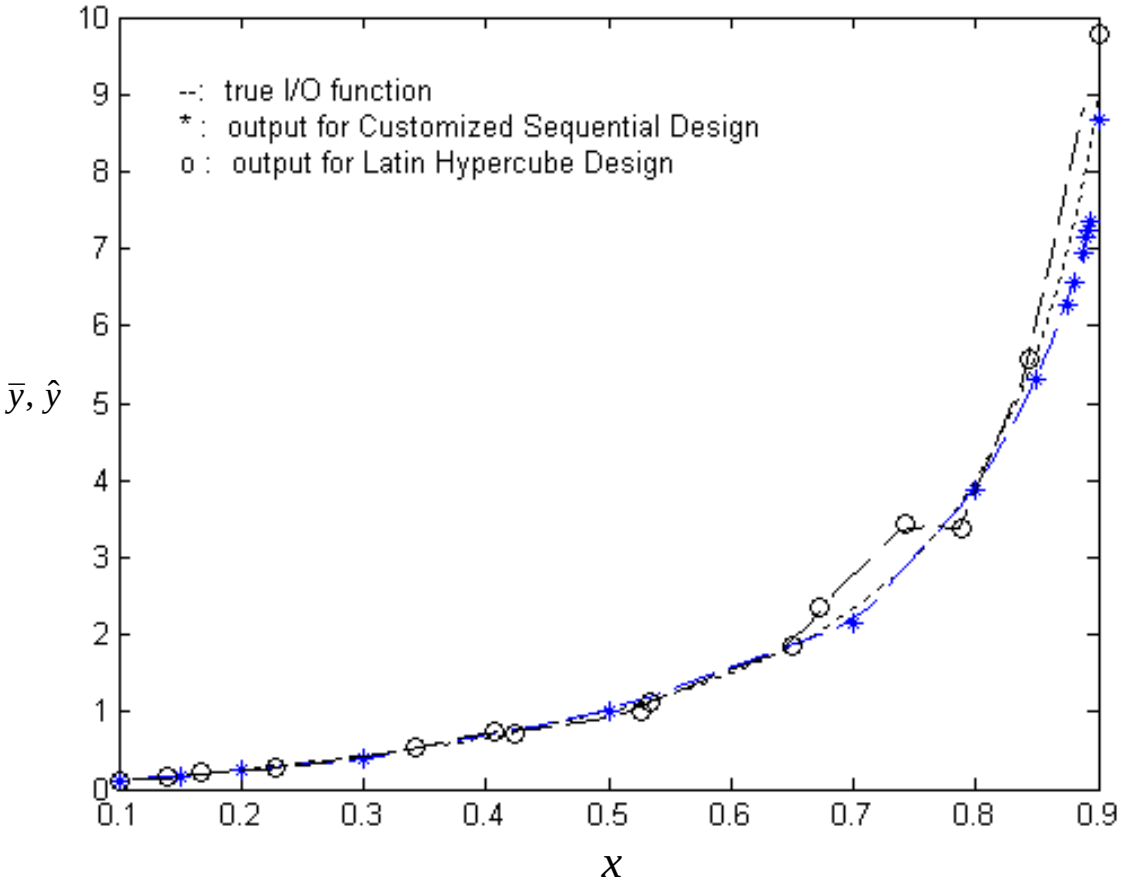

Figure 1. Two designs with 15 traffic rates x, their average simulation outputs $\bar{y}$, and the Kriging interpolations $\hat{y}$, for M/M/1

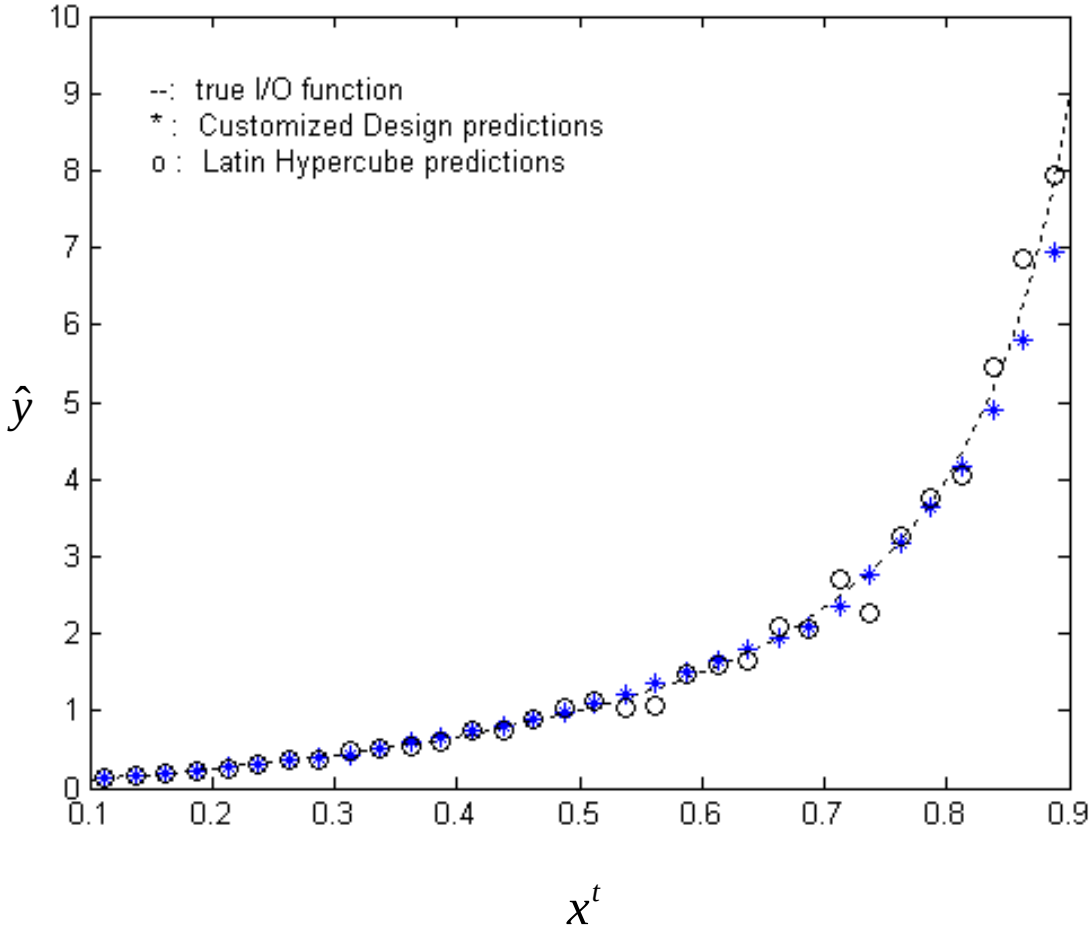

Figure 2. Two designs and their predictions $\hat{y}$ for the test set inputs x^t, for M/M/1

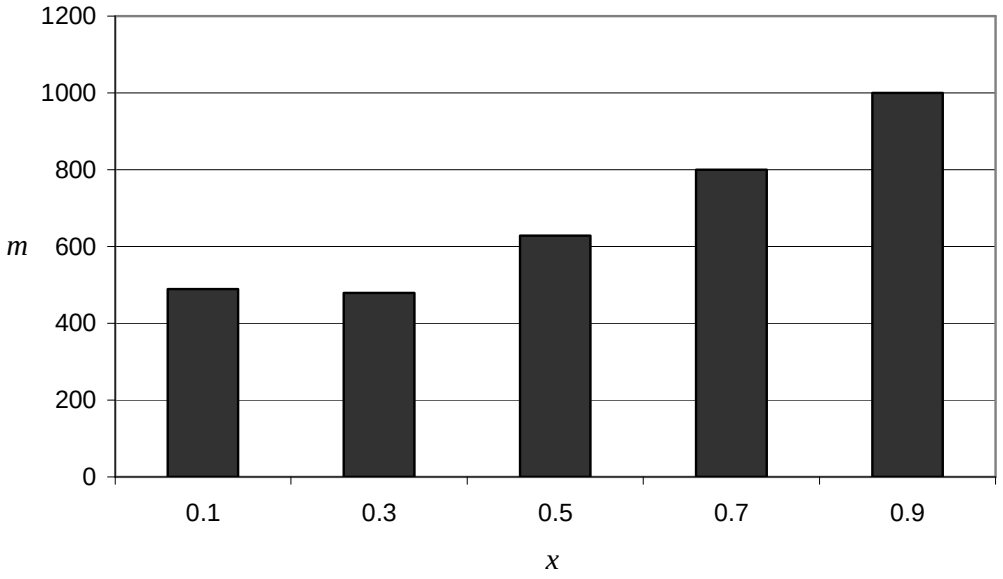

Figure 3. Number of cycles *m* per traffic rate *x* for M/M/1, if $\alpha = 0.05$ and $\delta = 0.15$

Case						a: seed = 10		b: seed = 12	
						CEIMSE	max MSE	CEIMSE	max MSE
1	$\alpha = 0.05$	$\delta = 0.15$	$B = 100$	$n = 15$	seq	0.067436	0.83412	0.038184	0.88275
					lhs	0.087018	1.86096	0.066435	0.89785
2				$n = 25$	seq	0.076384	0.83412	0.039545	0.88275
					lhs	0.092085	2.4571	0.057393	0.92206
3				$n = 100$	seq	0.068665	0.83412	0.039533	0.88275
					lhs	0.079271	0.52591	0.049694	0.32893
4	$\alpha = 0.05$	$\delta = 0.15$	$B = 50$	$n = 15$	seq	0.067437	0.83412	0.038185	0.88275
					lhs	0.083430	1.86096	0.067582	0.89785
5				$n = 25$	seq	0.076384	0.83412	0.039544	0.88275
					lhs	0.092085	2.4571	0.056007	0.92206
6	$\alpha = 0.05$	$\delta = 0.05$	$B = 100$	$n = 15$	seq	0.065212	0.83412	0.040773	0.60218
					lhs	0.072658	1.3592	0.041426	0.76348
7	$\alpha = 0.01$	$\delta = 0.15$	$B = 100$	$n = 15$	seq	0.064575	0.83412	0.047135	0.6769
					lhs	0.066912	1.0823	0.058562	0.76348

Table 1. Corrected Empirical Integrated Mean Squared Error (CEIMSE; see (19)) and maximum MSE (see (8), for M/M/1