Journal: JORS | Disk used: com OP: VVC/Scan: Deepa Ed: Viji
Article : ppl_jors_2601747 | Pages: 1–8 Despatch Date: 8/3/2004

Journal of the Operational Research Society (2004) 0, 1–8

www.palgrave-journals.com/jors

Application-driven sequential designs for simulation experiments: Kriging metamodelling

JPC Kleijnen* and WCM van Beers

Center for Economic Research (CentER), Tilburg University (UvT), Tilburg, The Netherlands

This paper proposes a novel method to select an experimental design for interpolation in simulation. Although the paper focuses on Kriging in deterministic simulation, the method also applies to other types of metamodels (besides Kriging), and to stochastic simulation. The paper focuses on simulations that require much computer time, so it is important to select a design with a small number of observations. The proposed method is therefore sequential. The novelty of the method is that it accounts for the specific input/output function of the particular simulation model at hand; that is, the method is application-driven or customized. This customization is achieved through cross-validation and jackknifing. The new method is tested through two academic applications, which demonstrate that the method indeed gives better results than either sequential designs based on an approximate Kriging prediction variance formula or designs with prefixed sample sizes.
Journal of the Operational Research Society (2004) **0,** 000–000. doi:10.1057/palgrave.jors.2601747

Keywords: simulation; statistics; stochastic; regression; methodology

Introduction

We are interested in *expensive simulations*; that is, we assume that a single simulation run takes 'much' computer time (say, its time is measured in days, not minutes). Therefore, we devise a method meant to minimize the number of simulation runs — that number is called the 'sample size' in statistics or the 'design size' or 'scheme size' in design of experiments (DOE).

We *tailor* our design to the actual simulation; that is, we do not derive a generic design such as a classic 2^{k-p} design or a Latin hypercube sampling (LHS) design. We can explain the differences between our designs on one hand and classic and LHS designs on the other hand, as follows.

Classic designs assume a simple 'metamodel' (also called approximate model, emulator, response surface, surrogate, etc). A *metamodel* is a model of an input/output (I/O) function. We denote the metamodel by $Y(\mathbf{x})$ where $\mathbf{x}$ denotes the k-dimensional vector of the k inputs — called 'factors' in classic DOE. In simulation, the true I/O function is implicitly defined by the simulation model itself (in real-life experiments, 'nature' defines this function). Classic 2^{k-p} designs of resolution III assume a first-order polynomial function (optimal resolution-III designs are orthogonal matrices, under various criteria). Central composite designs (CCD) assume a second-order polynomial function. See, for example, the well-known textbook of Box *et al*[1] or the recent textbook of Myers and Montgomery.[2]

LHS — much applied in Kriging — assumes I/O functions more complicated than classic designs do — but LHS does not specify a specific function for $Y(\mathbf{x})$. Instead, LHS focuses on the design space formed by the k-dimensional unit cube, defined by $0 \leqslant x_j \leqslant 1 (j = 1, \ldots, k)$ after standardizing (scaling) the inputs. LHS tries to sample that space according to some prior distribution for the inputs, such as independent uniform distributions on (or some nonuniform distribution in risk or uncertainty analysis); see McKay *et al*,[3] and also Koehler and Owen[4] and Kleijnen *et al*.[5]

Unlike LHS, we explicitly account for the I/O function; unlike, classic DOE we use a more realistic I/O function than a low-order polynomial. Therefore, we estimate the true I/O function through *cross-validation*; that is, we successively delete one of the I/O observations already simulated (for cross-validation see Stone;[6] for an update, see Meckesheimer *et al*[7] and Mertens[8]). In this way, we estimate the uncertainty of output at input combinations not yet observed. To measure this uncertainty, we use the *jackknifed* variance. For jackknifing see the classic article by Miller;[9] for an update, see again Meckesheimer *et al*[7] and Mertens.[8] We also compare our designs (based on cross-validation and jackknifing) to sequential designs based on a formula that approximates the variance of the Kriging predictor.

It turns out that our procedure concentrates on input combinations (design points, simulation scenarios) in subareas that have *more interesting* I/O behaviour. In our Example I, we spend most of our simulation time on the challenging 'explosive' part of a hyperbolic function (which

*Correspondence: JPC Kleijnen, Department of Information Systems and Management, Center for Economic Research (CentER), Tilburg University (Uvt), Postbox 90153, 5000 LE Tilburg, The Netherlands.
E-mail: kleijnen@uvt.nl
http://center.kub.nl/staff/kleijnen/

Gml : Ver 6.0
Template: Ver 1.0.1

may represent mean steady-state waiting time of single-server waiting systems). In Example II, we avoid spending much time on the relatively flat part of the fourth-degree polynomial I/O function with multiple local hills. (The reader may refer to Figures 3 and 6 discussed later.)

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

1. Sequential procedures are known to be more 'efficient'; that is, they require fewer observations than fixed-sample procedures; see the statistics literature, for example, Ghosh and Sen[10] and Park *et al.*[11]
2. Simulation experiments proceed sequentially (unless parallel computers are used).

Our application-driven sequential design (ADSD) does not provide tabulated designs; instead, we present a procedure for generating a sequential design for the actual (simulation) experiment.

Note that a different ADSD is developed by Sasena *et al.*[12] They, however, focus on optimization instead of sensitivity analysis (we think that optimization is more applied in engineering sciences than in management sciences, because the latter sciences involve softer performance criteria). Moreover, they use the 'generalized expected improvement function' assuming a Gaussian distribution, as proposed by Jones *et al.*[13] We, however, use distribution-free jackknifing and cross-validation for a set of candidate input combinations. Sasena *et al*[12] examine several criteria for selecting the next input combination to be simulated, including the 'maximum variance' criterion; the latter criterion is the one we use. (An alternative to their single, globally fitted Kriging metamodel for constrained optimization is a sequence of locally fitted first-order polynomials; see Angün *et al.*[14]) Related to Sasena *et al*[12] is Watson and Barnes.[15] More research is needed to compare our method with Sasena *et al*'s method (also see our final section, called 'Conclusions and further research').

The remainder of this paper is organized as follows. First, we summarize the basics of Kriging. Then we summarize DOE and Kriging. Subsequently, we explain our method, which uses cross-validation and jackknifing to select the next input combination to be simulated; this section also discusses sequentialization and stopping. Next, we demonstrate the procedure through two academic applications, which shows that our method gives better results than a design with a prefixed sample size; moreover, estimated Gaussian and linear correlation functions (variograms) — used in Kriging — give approximately the same results. The final section presents conclusions and topics for further research.

Kriging basics

Kriging is named after the South African mining engineer DG Krige. It is an interpolation method that predicts unknown values of a random function or random process; see Cressie's[16] classic Kriging textbook and Eq. (1) below. More precisely, a Kriging prediction is a weighted linear combination of all output values already observed. These weights depend on the distances between the location to be predicted and the locations already observed. Kriging assumes that *the closer the input data are, the more positively correlated the prediction errors are*. This assumption is modelled through the correlogram or the related variogram, discussed below.

Nowadays, Kriging is also popular in *deterministic simulation* (to model the performance of computer chips, television screens, etc); see Sacks *et al*'s[17] pioneering article, and — for an update — see Simpson *et al.*[18] Compared with linear regression analysis, Kriging has an important advantage in deterministic simulation: Kriging is an *exact interpolator*; that is, predicted values at observed input values are exactly equal to the observed (simulated) output values.

Kriging assumes the following *metamodel*:

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

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

As we mentioned above, the Kriging *predictor* for the unobserved input $\mathbf{x}_0$ — denoted by $\hat{Y}(\mathbf{x}_0)$ — is a weighted linear combination of all the (say) n observed output data:

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

with $\sum_{i=1}^{n} \lambda_i = 1$, $\lambda = (\lambda_1, \ldots, \lambda_n)'$ and $\mathbf{Y} = (y_1, \ldots, y_n)'$. To choose these weights, the 'best' linear unbiased estimator (BLUE) is derived: this estimator minimizes the mean-squared prediction error $\text{MSE}(\hat{Y}(\mathbf{x}_0)) = E((Y(\mathbf{x}_0) - \hat{Y}(\mathbf{x}_0))^2)$, with respect to λ. Obviously, this solution depends on the covariances, which may be characterized by the *variogram*, defined as $2\gamma(\mathbf{h}) = var(Y(\mathbf{x}+\mathbf{h}) - Y(\mathbf{x}))$. (We follow Cressie,[16] who uses variograms, whereas Sacks *et al.*[17] use correlation functions; also see our discussion on the estimation of variograms in the section called 'Two examples'.) An example variogram is given in Figure 1.

It can be proven that the *optimal* weights in (2) are

$$\lambda' = \left(\gamma + \mathbf{1}\frac{1 - \mathbf{1}'\Gamma^{-1}\gamma}{\mathbf{1}'\Gamma^{-1}\mathbf{1}}\right)' \Gamma^{-1} \quad (3)$$

where γ is the vector of (co)variances $\gamma(\mathbf{x}_0 - \mathbf{x}_1), \ldots, \gamma(\mathbf{x}_0 - \mathbf{x}_n))'$; $\boldsymbol{\Gamma}$ is the $n \times n$ matrix whose (i, j)th element is $\gamma(\mathbf{x}_i - \mathbf{x}_j)$; $\mathbf{1} = (1, \ldots, 1)'$ is the vector of ones. We point out

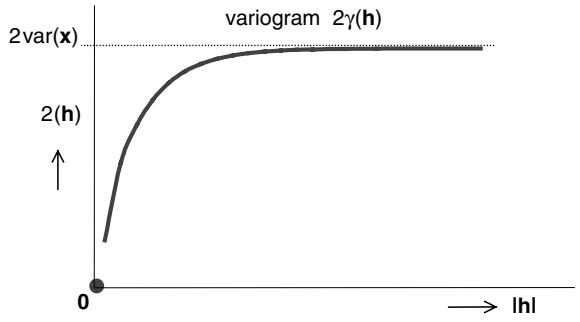

Figure 1 An example variogram.

that the weights in (3) vary with the prediction point, whereas regression analysis uses the same estimated meta-model for all prediction points.

As the (co)variances in (3) are unknown, they are based on the estimated variogram. If the *random* character of the resulting estimated optimal weights $\hat{\lambda}$ is ignored, then the variance of the resulting linear estimator at a fixed point $\boldsymbol{x}_0$ is

$$\sigma_k^2(\mathbf{x}_0 \left| \hat{\lambda} = \lambda) = 2 \times \sum_i^n \lambda_i \gamma(\mathbf{x}_0 - \mathbf{x}_i) - \sum_i^n \sum_j^n \lambda_i \lambda_j \gamma(\mathbf{x}_i - \mathbf{x}_j) \quad (4)$$

see Cressie[16] (p 122), who does not explicitly mention the conditional character of (4).

Further details on Kriging are provided by Cressie[16] and an update by Van Beers and Kleijnen.[19]

DOE and Kriging

A *design* is a set of (say) n combinations of the k factor values. These combinations are usually bounded by 'box' constraints: $a_j \leqslant x_j \leqslant b_j$, 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 (also see the Introduction).

Our goal is to find a design — for Kriging predictions within H — with the *smallest size* that satisfies a certain criterion. The literature proposed several criteria: see Sacks *et al*[17] (p 414). Most of these criteria are based on the mean squared prediction error, $\text{MSE}(\hat{Y}(\mathbf{x})) = E(\hat{Y}(\mathbf{x}) - Y(\mathbf{x}))^2$ where the predictor $\hat{Y}(\mathbf{x})$ follows from (2) and the true output $Y(\mathbf{x})$ was defined in (1). (An alternative considers $100(1-\alpha)\%$ prediction regions for $y(\mathbf{x})$ and inter-quantile ranges for $\hat{y}(\mathbf{x})$; see Cressie,[16] p 108.) However, most progress has been made through the integrated mean squared error (IMSE); see Bates *et al*:[20] choose the design that minimizes

$$\text{IMSE} = \int_H \text{MSE}(\hat{Y}(\mathbf{x}))\phi(\mathbf{x})\mathrm{d}x \quad (5)$$

for a given weight function $\phi(\mathbf{x})$.

To validate the design, Sacks *et al*.[17] (p 416) compare the predictions with the known true values in a *test set* of size (say) m. They assume $\phi(\mathbf{x})$ to be uniform, so IMSE in (5) can be estimated by the empirical integrated mean squared error (EIMSE):

$$\text{EIMSE} = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i(\mathbf{x}) - y_i(\mathbf{x}))^2 \quad (6)$$

Note that criteria such as (5) are more appropriate in sensitivity analysis than in simulation optimization; see Sasena *et al*[12] and also Kleijnen and Sargent[21] and Kleijnen.[22]

Application-driven sequential design

Pilot input combinations

We start with a *pilot design* of size (say) n_0. To select n_0 *specific* points, we notice that Kriging gives very bad predictions in case of *extrapolation* (ie, predictions outside the convex hull of the observations obtained so far). Indeed, in our examples we find very bad results (not displayed). Therefore, we select the 2^k vertices of H as a subset of the pilot design. In our two examples with a *single* input ($k = 1$), this choice implies that one input value is the minimum and one is the maximum of the input's range; see Figure 2 (other parts of this figure will be explained below, in next subsections).

Besides these 2^k vertices, we must select some more input combinations to *estimate the variogram*. Like Cressie[16] we assume either a Gaussian variogram

$$\gamma(h) = c_0 + c_1(1 - \exp(-h/a)) \quad (7)$$

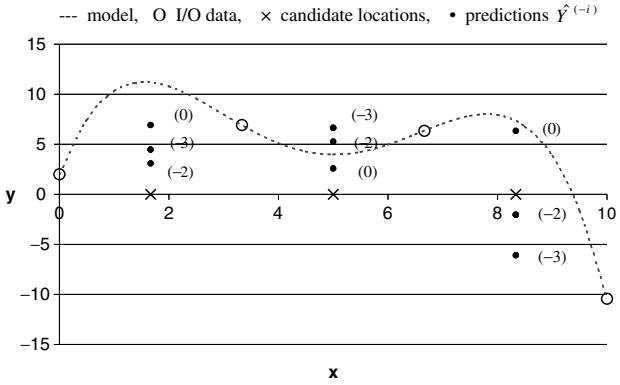

Figure 2 Fourth-order polynomial example, including four pilot observations and three candidate inputs with predictions based on cross-validation, where $(-i)$ denotes which observation i is dropped in the cross-validation.

or a linear variogram

$$\gamma(h) = c_0 + ch \tag{8}$$

Obviously, estimation of variogram (7) requires at least three different values of h (for example, the values 0, $\frac{1}{2}$, 1); thus at least three different I/O combinations. Moreover — as we shall see — our approach uses cross-validation, which implies that we drop one of the n_0 observations and re-estimate the variogram; that is, cross-validation necessitates one extra I/O combination.

In practice, we may select a 'small' set of additional observations — besides the 2^k corner points — using a standard *space-filling design*, which ensures that no two design points are too close to each other. More specifically, we propose a *maximin* design, which packs all design points in hyper spheres with maximum radius; see Koehler and Owen[4] (p 288). In our examples, we take — besides the two end points of the factor's range — two additional points. The latter points we place such that all four observed points are equidistant; see again Figure 2. (Future research may investigate alternative sizes n_0 and components $\boldsymbol{x}$.)

Candidate input combinations

After selecting and actually simulating a pilot design, we choose additional input combinations — accounting for the particular simulation model at hand. Since we do not know the I/O function of this simulation model, we choose (say) c *candidate points* — without actually running any expensive simulations for these candidates (as we shall see in next subsection).

First, we must select a *value* for c. In Figure 2, we select three candidate input values (had we taken more candidates, then we would have to perform more Kriging calculations; in general, the latter calculations are small compared with the 'expensive' simulation computations).

Next, we must select c *specific* candidates. Again, we use a space-filling design (as we did for the pilot sample). In Figure 2, we select the three candidates *halfway* between the four input values already observed. (Future research may investigate how to use a space-filling design to select candidates, ignoring candidates that are too close to the points already observed. In practice, LHS designs are attractive since they are so simple: LHS is part of spreadsheet add-ons such as @Risk.)

Cross-validation

To select a 'winning' candidate for actual (expensive) simulation, we estimate the variance of the predicted output at each candidate input — without any actual simulation. Therefore, we use cross—validation and jackknifing, as follows.

Given a set of observed I/O data (x_i, y_i) with $i = 1, \ldots, n$ (initially, $n = n_0$), we eliminate observation i and obtain the *cross-validation* sample (with only $n-1$ observations):

$$\begin{aligned} S^{(-i)} = \{&(x_1, y_1), (x_2, y_2), \ldots, (x_{i-1}, y_{i-1}), \\ &(x_{i+1}, y_{i+1}), \ldots, (x_n, y_n)\}. \end{aligned} \tag{9}$$

From the sample in Eq. (9), we could compute the Kriging prediction for the output for each candidate. However, to avoid extrapolation (see previous section 'Pilot input combinations'), we do not eliminate the observations at the vertices: of the cross-validation sample in (9) we use only (say) n_c observations. The predictions are analogous to (2) replacing n by n_c; in case of $k = 1$ we take $n_c = n_0 - 1$. Obviously, we must re-estimate the optimal weights in (2), using (3) (also see the 'binning' discussion at the end of next subsection). Figure 2 shows the $n_c = n_0 - 1 = 3$ Kriging predictions (say) $\hat{Y}^{(-i)}$ after deleting observation i as in (9), for each of the $c = 3$ candidates.

Figure 2 suggests that it is most difficult to predict the output at the candidate point $x = 8.33$. To quantify this prediction uncertainty, we use jackknifing.

Jackknifing

First, we calculate the jackknife's *pseudo-value* for candidate j, which is defined as the following weighted average of the original and the cross-validation predictors:

$$\begin{aligned} \tilde{y}_{j;i} = &n_c \times \hat{Y}_j^{(-0)} - (n_c - 1) \times \hat{Y}_j^{(-i)} \\ &\text{with } j = 1, \ldots, c \text{ and } i = 1, \ldots, n_c \end{aligned} \tag{10}$$

where $\hat{Y}_j^{(-0)}$ is the original Kriging prediction for candidate input j based on the complete set of observations (zero observations eliminated: see the superscript -0).

From the pseudo-values in (10), we estimate the *jackknife variance* for candidate j:

$$\tilde{s}_j^2 = \frac{1}{n_c(n_c - 1)} \sum_{i=1}^{n_c} (\tilde{y}_{j;i} - \overline{\tilde{y}}_j)^2 \text{ with } \overline{\tilde{y}}_j = \frac{1}{n_c} \sum_{i=1}^{n_c} \tilde{y}_{j;i} \tag{11}$$

Note that we also experimented with other measures of variability, for example, the 90% interquantile; all these measures gave the same type of design.

Finally, to select the *winning* candidate (say) w for actual simulation, we find the maximum of the jackknife variances in (11):

$$w = \arg\left(\max_j \{\tilde{s}_j^2\} \right) \tag{12}$$

Note that a *candidate* location close to a *deleted* observation lies relatively far away from the remaining observations. Hence, such a candidate is less correlated with its neighbouring points. Consequently, its Kriging predictor becomes rather uncertain. However, this phenomenon holds for each deleted observation.

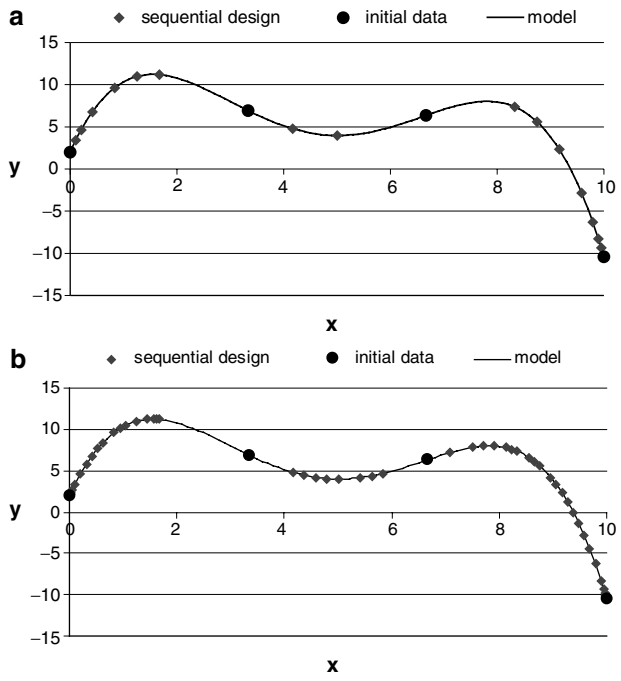

Figure 3 Figure 2 continued with (a) $n = 19$ observations and (b) $n = 54$ observations.

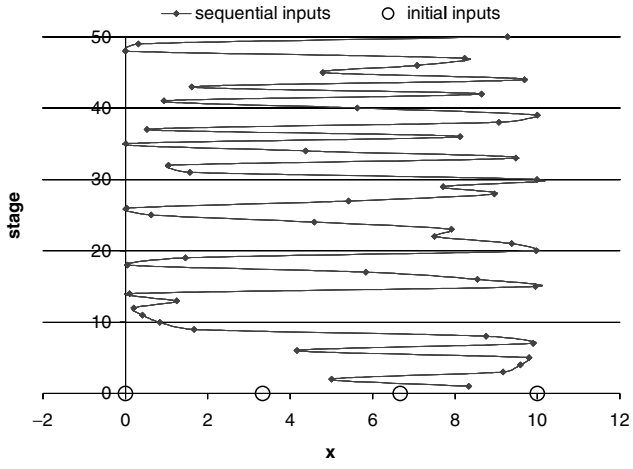

Figure 4 Dynamics of sequential sampling for Example I.

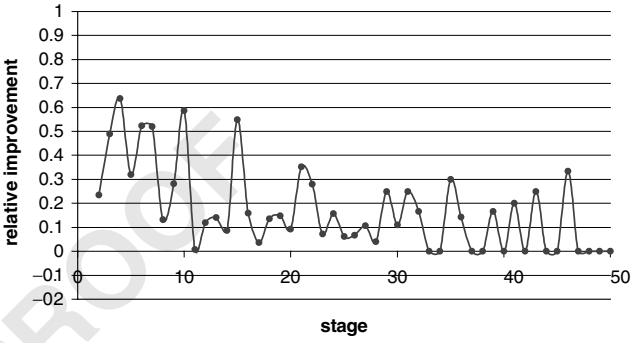

Figure 5 Successive relative improvements for 50 observations in hyperbole example.

Note further that to reduce the computer time needed by our procedure (not by the simulation itself), we estimate the variogram from *binned* distances: for n inputs, we classify the $n(n-1)/2$ possible distances h in (say) $n_b < n$ equally sized intervals or 'bins'. These intervals should be as small as possible to retain spatial resolution, yet large enough to stabilize the variogram estimator. Journel and Huijbregts[23] recommend at least 30 distinct pairs in each interval. For the n_b midpoints of these intervals, we calculate the average squared difference to estimate the variogram; see Cressie[16] (p 69). In our examples we use $n_b = 15$.

Sequentialization

Once we have simulated the 'winning' candidate selected through (12), we add the new observation to the set of observations; see S in (9) — now with superscript (-0) and with $n+1$ members.

Next, we choose a new set of candidates with respect to this augmented set. For example, in Figure 2 we add as new candidates $x = 1.67$, 5, 7.5 and 9.17; these candidates are not shown in Figure 2, but the winning candidate is shown as part of Figure 3.

The 'dynamics' of our procedure is demonstrated by Figure 4, which shows the *order* in which input values are selected — in a total sample size $n = 50$.

Stopping rule

To stop our sequential procedure, we measure the successive relative improvement (SRI) after n observations:

$$\mathrm{SRI}_n = |\max_j \{\tilde{s}_j^2\}_n - \max_j \{\tilde{s}_j^2\}_{n-1}| / \max_j \{\tilde{s}_j^2\}_{n-1} \qquad (13)$$

where $\max_j \{\tilde{s}_j^2\}_n$ denotes the maximum jackknife variance (see (12)) after n observations. Figure 5 shows SRI for up to $n = 50$ in Example I (detailed in next subsection). There are no essential changes in (13) beyond $n = 15$. In the literature (including Sasena *et al*[12] and Jones *et al*[13]), we did not find an appealing stopping criterion for our sequential design; future research may be needed.

We *stop* our sequential procedure as soon as we find no 'substantial' reduction for SRI. However, SRI may fluctuate greatly in the first stages, so we might stop *prematurely*. To avoid such stopping, we select a minimum value (say) $n_{\min}$ so that the complete design contains $n = n_0 + n_{\min}$ observations. Figure 3(a) used $n_{\min} = 15$, whereas Figure 3(b) used $n_{\min} = 50$ (Figure 2 is the part of Figure 3 that corresponds with $n = 4$).

In practice — as Kleijnen *et al*[5] point out —simulation experiments may stop prematurely (eg, the computer may break down). Our procedure then still gives useful information.

Two examples

Example I: a hyperbolic I/O function

Consider the following hyperbole:

$$y = \frac{x}{1-x} \quad \text{with} \quad 0 < x < 1 \tag{14}$$

We are interested in this example, because y in (14) equal the expected waiting time in the steady state of a single-server system with Markovian (Poisson) arrival and service times (denoted by M/M/1). This system has a single input parameter, namely the traffic load x, which is the ratio of the arrival rate and the service rate. This system is a building block in many realistic discrete-event simulation models; see Law and Kelton[24] (p 12) and also Van Beers and Kleijnen.[19]

When applying our approach to (14), we decided to select a pilot sample size $n_0 = 4$ and a minimum sample size value $n_{\min} = 10$. We stop the sequential procedure as soon as the SRI in (13) drops below 5%; this results in a total sample size $n = 19$. Also see Figure 6(a). Replacing 5 by 1% gives $n = 36$; see Figure 6(b).

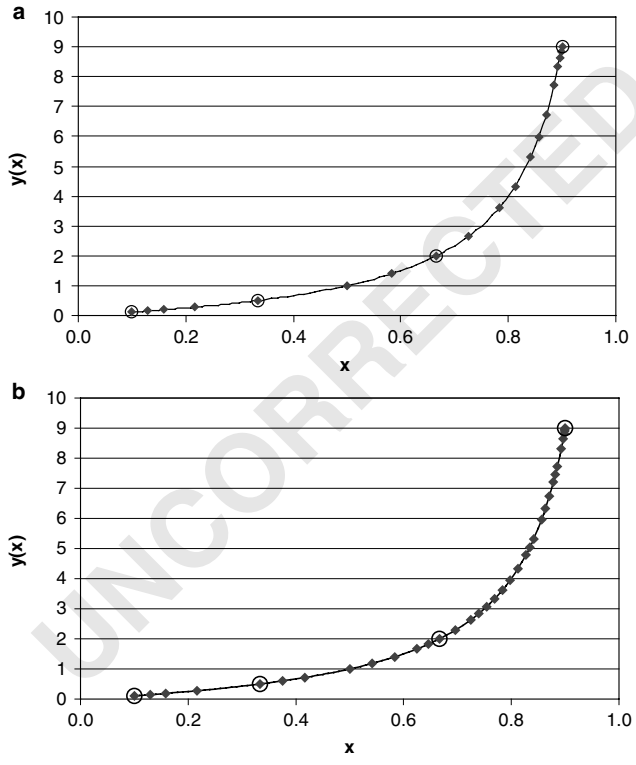

Figure 6 Hyperbole example, including four pilot observations with (a) $n = 19$ observations and (b) $n = 36$ observations.

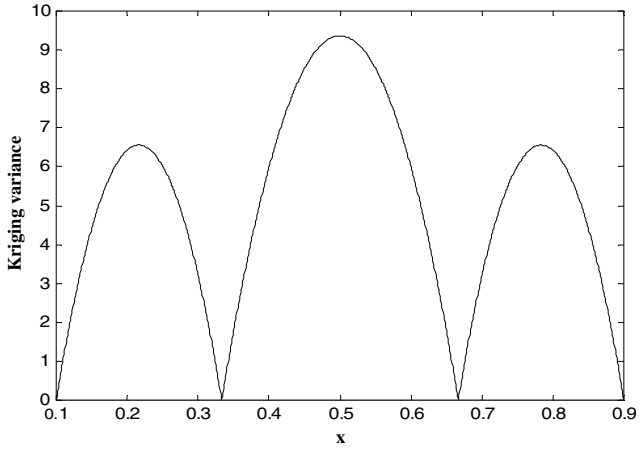

Figure 7 Approximate Kriging variance in initial design.

Figure 6 demonstrates that our final design selects relatively few input values in the area that generates an approximately linear I/O function, whereas it selects many input values in the exploding part (where x approaches one).

We think that our design is intuitively appealing — but we also use a *test set* to quantify its performance. In this test, we compare our design with two alternative design types of the same size ($n = 19$ or 36):

(i) A *sequential* design based on the *approximate* Kriging variance formula (4). We then select as the next point the input value that maximizes this variance (we do not need to specify candidate points); see Figure 7. The figure illustrates that this approach selects as the next point the input farthest away from the old inputs, namely $x = 0.5$ (also see Goovaerts's statement on http://www.sph.umich.edu/geomed/mods/geostats_lite/lec/krigvariance.html). This results in a final design that spreads all its points evenly across the experimental area (so it resembles the next design).

(ii) A *single-stage LHS design*: LHS divides the total range of the input variable into n mutually exclusive and exhaustive intervals of equal length. Within each interval, LHS samples a uniformly distributed value. To estimate the resulting variability, we decided to obtain 10 LHS samples, from which we estimate the mean and the standard deviation (standard error).

From the n *o*bservations per design we compute the Kriging predictors for the 32 true test values, and calculate the squared error per test value. From the 32 values we compute the average — see EIMSE in (6), which corresponds with the L_2 norm — and the maximum or L_∞ norm. We find substantially better results for our designs; see Table 1.

Table 1 IMSE of three design type for hyperbole (Example I)

	ADSD		*Krig Var*		*LHS*	
n	*EIMSE*	L_∞	*EIMSE*	L_∞	*EIMSE*	L_∞
19	8.90×10^{-4}	0.0759	80.08×10^{-4}	0.3460	61.4×10^{-4} (48.1×10^{-4})	0.3559 (0.1740)
36	1.19×10^{-4}	0.0303	8.11×10^{-4}	0.1501	2.76×10^{-4} (0.98×10^{-4})	0.0791 (0.0185)

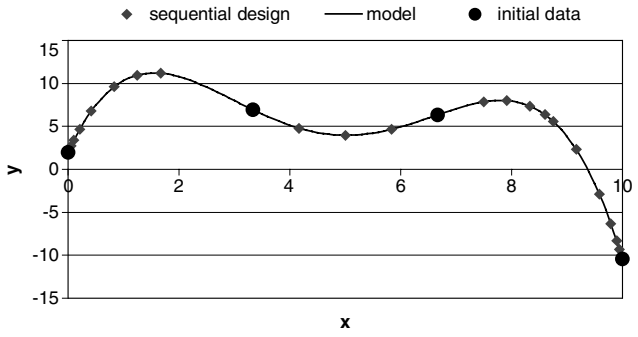

Figure 8 Final design for fourth-order polynomial example with RSI < 1% and $n = 24$.

Example II: a fourth-order polynomial I/O function

As Van Beers and Kleijnen[19] did, we consider

$$y = -0.0579x^4 + 1.11x^3 - 6.845x^2 + 14.1071x + 2 \quad (15)$$

which is a multimodal function; see again Figure 2.

For our design, we select $n_0 = 4$, $n_{\min} = 10$, and an SRI smaller than 5%. This gives a sequential design with 18 observations. An SRI smaller than 1% gives a final (sequential) design with 24 observations (Example I resulted in 36 observations).

Figure 8 demonstrates that our final design selects relative few input values in the area that generates an approximately linear I/O function, whereas it selects many input values near the edges, where the function changes much.

We again compare our design with the two alternative designs discussed above. We find substantially better results for our designs; see Table 2.

Note that we focus on sensitivity analysis, not optimization. For example, our method selects input values — not only near the 'top' — but also near the 'bottom' of (15). If we were searching for a maximum, we would adapt our procedure such that it would not collect data near an obvious minimum.

Estimated variograms: Gaussian versus linear

We also investigate the influence of the assumed variogram, namely a Gaussian variogram and a linear variogram; see (7) and (8). We use a single-stage design with 21 observations. We use ordinary least squares for these estimators (whereas Sacks *et al*[17] assume a Gaussian correlation function and use maximum likelihood estimation, which takes much more computer time and may involve numerical problems).

The Gaussian and the linear variograms result in two designs that look very similar, for both Examples I and II. More precisely, when using a test set of nine equidistant input values, Kriging predictions based on a Gaussian variogram give an EIMSE of 0.3702, whereas a linear variogram gives 0.3680 for Example I. Analogously, Example II gives 0.0497 and 0.0482. So the Gaussian and linear variograms give similar values for EIMSE. The linear variogram, however, is simpler: no data transformation is needed.

Conclusions and further research

To avoid expensive simulation runs, we propose cross-validation and jackknifing to estimate the variances of the outputs for *candidate* input combinations. We actually simulate only the candidate with the *highest* estimated variance. This procedure we apply *sequentially*.

Table 2 IMSE for three design types for fourth-degree polynomial

	ADSD		*Krig Var*		*LHS*	
n	*EIMSE*	L_∞	*EIMSE*	L_∞	*EIMSE*	L_∞
18	0.1741	1.0470	0.5793	0.6718	0.5855 (0.5574)	3.3011 (1.9706)
24	0.0121	0.2503	0.2690	0.5133	0.2473 (0.2112)	2.1212 (1.3837)

Our two examples show that our procedure simulates relatively many input combinations in those sub-areas that have interesting I/O behaviour. Our design gives smaller prediction errors than either sequential designs based on the approximate variance formula in (4) or single-stage designs do.

In future research, we may extend our approach to

1. alternative *pilot-sample* sizes n_0 with alternative space-filling input combinations $\boldsymbol{x}$ (Jones *et al*[13], p 21, propose $n_0 = 10k$ and an adjusted LHS design);
2. alternative space-filling designs for the selection of *candidate* input combinations, ignoring candidates that are too close to the points already observed in any preceding stages (such an alternative design may be a nearly orthogonal LHS design; see Kleijnen *et al*[5]);
3. a *stopping criterion* for our sequential design;
4. *multiple* inputs ($k > 1$);
5. *realistic* simulation models (instead of our Examples I and II);
6. comparison of our approach with *Sasena et al*'s[12] approach;
7. *stochastic* simulation models (focus of our current research);
8. *other metamodels*, such as linear regression models (see Kleijnen and Sargent[21]) and neural nets (see Simpson *et al*.[25])

Acknowledgements—Bert Bettonvil (Tilburg University) and Paul Switzer (Stanford University) provided very useful comments on an earlier version.

References

1 Box GEP, Hunter WG and Hunter JS (1978). *Statistics for Experimenters: An Introduction to Design, Data Analysis and Model Building*. Publisher: John Wiley & Sons, Inc.: New York.

2 Myers RH and Montgomery DC (2002). *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. 2nd edn Wiley: New York.

3 McKay MD, Beckman RJ and Conover WJ (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**(2): 239–245 (reprinted in 2000: Technometrics, 42 (1): 55–61.

Q1 4 Koehler JR and Owen AB (1996). Computer experiments. In: Ghosh S and Rao CR (eds) *Handbook of Statistics* Vol **13**, pp 261–308.

5 Kleijnen JPC, Sanchez SM, Lucas TW and Cioppa TM (2002). A user's guide to the brave new world of designing simulation experiments. Working Paper (preprint: http://center.kub.nl/staff/kleijnen/papers.html).

6 Stone M (1974). Cross-validatory choice and assessment of statistical predictions. *J R Stat Soc, Ser B* **36**(2): 111–147.

7 Meckesheimer M, Barton RR, Simpson TW and Booker AJ (2002). Computationally inexpensive metamodel assessment strategies. *AIAA J* **40**(10): 2053–2060.

8 Mertens BJA (2001). Downdating: interdisciplinary research between statistics and computing. *Statist Neerlandica* **55**(3): 358–366.

9 Miller RG (1974). The jackknife — a review. *Biometrika* **61**: 1–15.

10 Ghosh BK and Sen PK (eds) (1991). *Handbook of Sequential Analysis*. Marcel Dekker, Inc.: New York.

11 Park S *et al* (2002). D-optimal sequential experiments for generating a simulation-based cycle time-throughput curve. *Opns Res* **50**(6): 981–990.

12 Sasena MJ, Papalambros P and Goovaerts P (2002). Exploration of metamodeling sampling criteria for constrained global optimization. *Eng Optim* **34**(3): 263–278.

13 Jones DR, Schonlau M and Welch WJ (1998). Efficient global optimization of expensive black-box functions. *J Global Optim* **13**: 455–492.

14 Angün ED, den Hertog Gürkan G and Kleijnen JPC (2002). Response surface methodology revisited. In: Yücesan E, Chen CH, Snowdon JL and Charnes JM (eds.) *Proceedings of the 2002 Winter Simulation Conference*, pp. 377–383. Q2

15 Watson AG and Barnes RJ (1995). Infill sampling criteria to locate extremes. *Math Geol* **27**(5): 589–608.

16 Cressie NAC (1993). *Statistics for Spatial Data*. Wiley: New York.

17 Sacks J, Welch WJ, Mitchell TJ and Wynn HP (1989). Design and analysis of computer experiments. *Statist Sci* **4**(4): 409–435.

18 Simpson TW, Mauery TM, Korte JJ and Mistree F (2001a). Kriging metamodels for global approximation in simulation-based multidisciplinary design optimization. *AIAA J* **39**(12): 2233–2241.

19 Van Beers WCM and Kleijnen JPC (2003). Kriging for interpolation in random simulation. *J Opl Res Soc*. accepted for publication. Q3

20 Bates RA, Buck RJ, Riccomagno E and Wynn HP (1996). Experimental design and observation for large systems. *R Stat Soc* **58**(1): 77–94.

21 Kleijnen JPC and Sargent RG (2000). A methodology for the fitting and validation of metamodels in simulation. *Eur J Opl Res* **120**(1): 14–29.

22 Kleijnen JPC (1998). Experimental design for sensitivity analysis, optimization, and validation of simulation models. In: Banks J (ed). *Handbook of Simulation*. Wiley: New York, pp 173–223 Chapter 6.

23 Journel AG and Huijbregts CJ (1978). *Mining Geostatistics*. Academic Press: London.

24 Law AM and Kelton WD (2000). *Simulation Modeling and Analysis*. 3rd edn. McGraw-Hill: Boston, MA.

25 Simpson TW, Peplinski J, Koch PN and Allen JK (2001b). Metamodels for computer-based engineering design: survey and recommendation. *Eng Comput* **17**(2): 129–150.

Received May 2003;
accepted February 2004