KRIGING METAMODELING IN CONSTRAINED SIMULATION OPTIMIZATION: AN EXPLORATIVE STUDY

William E. Biles

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

Inneke van Nieuwenhuyse

Department of Industrial Engineering
University of Louisville
Louisville, K.Y. 40292, U.S.A.

Department of Information Management
Tilburg University
5000 LE Tilburg, THE NETHERLANDS

Department of Decision Sciences
and Information Management
Catholic University of Leuven
Leuven, BELGIUM

ABSTRACT

This paper describes two experiments exploring the potential of the Kriging methodology for constrained simulation optimization. Both experiments study an (s, S) inventory system with the objective of finding the optimal values of s and S. The goal function and constraints in these two experiments differ, as does the approach to determine the optimum combination predicted by the Kriging model. The results of these experiments indicate that Kriging offers opportunities for solving constrained optimization problems in stochastic simulation; future research will focus on further refining the methodology.

1 INTRODUCTION

A *metamodel*, also called a *response surface*, is an approximation of an Input/Output (I/O) function that is defined by an underlying simulation model (see Kleijnen, 2007). The metamodel is the surrogate for the simulation model of the real-world system. This metamodel is used to guide experimentation and analysis. Experimentation with the actual system is far too costly and time consuming, so that computer-based experimentation, or *simulation*, is preferred.

Most metamodeling studies focus on low-order polynomial regression using factorial-based designs. The *central composite design* (CCD) is a popular experimental design for estimating a quadratic regression in a local, unimodal region of the design space. The difficulty with factorial-based designs such as the CCD is that they are *not space-filling*, as illustrated in Figure 1. In this figure, the "sum" values at the right end of each row and the bottom of each column show this situation quite clearly; that is, the many zeros among these "sum' values demonstrate the non-space filling characteristic of CCD designs. The presence of "sum" values > 1 signify yet another undesirable feature of CCD designs from the standpoint of Kriging; that is, CCD designs are *collapsing*, and as such are undesirable for the application of Kriging.

S \ s	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	Sum
40								1								1
45																0
50			1										1			2
55																0
60																0
65																0
70																0
75	1							5							1	7
80																0
85																0
90																0
95																0
100			1										1			2
105																0
110								1								1
Sum	1	0	2	0	0	0	0	7	0	0	0	0	2	0	1	13

Figure 1: A central composite design for two inputs

An alternative to factorial-based designs such as the CCD are *space-filling designs* such as the Latin Hypercube Design or LHD (McKay et al. 1979). These designs have especially favorable characteristics when applied in a constrained optimization experimental setting in that there is a high probability that one or more design points will not only fall within the feasible region (which is not known *a priori*), but will actually fall close to a constrained optimum solution. LHD designs are also especially well suited for Kriging in that they can be made to cover the design space in such a way that design points are almost equidistant from one another in k-dimensional space where k denotes the number of factors (also called inputs). Figure 2 illustrates a *non-collapsing, space-filling* LHD, as contrasted to the CCD in Figure 1; the fact that in Figure 2 all the "sum" values are 1 is an indication of the non-collapsing, space-filling features of LHD designs. Designs such as illustrated in Figure 2 are preferred when applying

Kriging to simulation experimentation. Popular software for obtaining a LHD is Crystal Ball, @Risk, and Risk Solver; also see the references in (Kleijnen 2007).

s	S	40	42	44	46	48	50	52	54	56	58	60	62	64	66	68	70	72	74	76	78	Sum
20						1																1
21									1													1
22											1											1
23																		1				1
24		1																				1
25						1																1
26									1													1
27												1										1
28										1												1
29			1																			1
30							1															1
31											1											1
32														1								1
33											1											1
34				1																		1
35							1															1
36													1									1
37																1						1
38															1							1
39				1																		1
Sum		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	20

Figure 2: A latin hypercube design for two inputs

2 KRIGING: BASICS

Kriging is an interpolation method that predicts unknown values of a random function. The simplest Kriging model is displayed in (1):

$$\hat{y}(x) = \beta + z(x) \tag{1}$$

where $\hat{y}(x)$ denotes the Kriging predictor for input or factor combination (point) x, β a constant, and $z(x)$ a covariance-stationary process; see, for example, Cressie (1993) and Wackernagel (2003).

A Kriging prediction is a weighted linear combination of all output values already observed:

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

The weights $\lambda = (\lambda_1, \dots, \lambda_n)'$ in (2) depend on the distances between the input to be predicted $\mathbf{x}_0$ and the inputs already observed $\mathbf{x}_i$. Kriging assumes that *the closer the input data are, the more positively correlated the prediction errors are*. This assumption is modeled through the correlogram (or the related variogram). In simulation, a popular class of correlograms is

$$\rho(\mathbf{h}) = \prod_{j=1}^{k} \exp(-\theta_j \cdot | h_j |^{p_j}) \tag{3}$$

where k denotes the number of inputs, $\mathbf{h} = (h_1, \dots, h_k)'$ the distance vector between two inputs, say $\mathbf{x}_i$ and $\mathbf{x}_{i'}$, θ_j the importance of input j; that is, the higher θ_j is, the less effect input j has, and p_j the smoothness of the correlogram function. Often, these powers p_j are chosen as $p_j = p = 2$. Then, the resulting correlogram is the infinitely differentiable so-called "Gaussian" correlation function.

The criterion to select the weights λ is mean-squared prediction error σ_e^2 defined as

$$\sigma_e^2 = E\left((Y(\mathbf{x}_0) - \hat{Y}(\mathbf{x}_0))^2\right). \tag{4}$$

Minimizing (4) under the constraint (2) gives the optimal weights

$$\lambda' = \left(\gamma + 1\frac{1 - 1'\Gamma^{-1}\gamma}{1'\Gamma^{-1}1}\right)'\Gamma^{-1} \tag{5}$$

where γ denotes the vector of (co)variances $(\gamma(\mathbf{x}_0 - \mathbf{x}_1), \dots, \gamma(\mathbf{x}_0 - \mathbf{x}_n))'$, Γ denotes the $n \times n$ matrix whose $(i, j)^{\text{th}}$ element is $\gamma(\mathbf{x}_i - \mathbf{x}_j)$, $1 = (1, \dots, 1)'$ is the vector of ones; also see Cressie (1993, p. 122). Note that some of the weights λ_i may be negative.

The optimal weights (5) give the minimal mean-squared prediction error (also see (Cressie 1993, p. 122))

$$\sigma_e^2 = \gamma'\Gamma^{-1}\gamma - \frac{(1'\Gamma^{-1}\gamma - 1)^2}{1'\Gamma^{-1}1}. \tag{6}$$

Actually, $\gamma(\mathbf{h})$ used in (5) and (6), is *unknown*. The usual *estimator* of this covariance is

$$2\hat{\gamma}(\mathbf{h}) = \frac{1}{|N(\mathbf{h})|} \sum_{N(\mathbf{h})} \left(Y(\mathbf{x}_i) - Y(\mathbf{x}_j)\right)^2 \tag{7}$$

where $|N(\mathbf{h})|$ denotes the number of distinct pairs in $N(\mathbf{h}) = \{(\mathbf{x}_i, \mathbf{x}_j) : \mathbf{x}_i - \mathbf{x}_j = \mathbf{h}; i, j = 1, \dots, n\}$; see Matheron (1962). These estimates imply estimates of the corresponding correlations, so the parameters θ_j and p_j in (3) can be fitted. For this fitting, standard Kriging software uses Maximum Likelihood Estimation (MLE); Van Beers and Kleijnen (2003), however, use Weighted Least Squares (WLS) estimation for a linear correlogram function. These estimated covariances are substituted into (5) to estimate the optimal weights. The statistical complications

arising from this estimation are discussed in Den Hertog, Kleijnen, and Siem (2006).

3 KRIGING IN CONSTRAINED OPTIMIZATION

Kleijnen and van Beers (2004) applied Kriging for sensitivity analysis of stochastic simulation with a single output. Now we explore the use of Kriging in constrained optimization.

We studied two experiments with an (s, S) inventory system. In such a system, a replenishment order is placed as soon as the inventory position (= on-hand inventory + outstanding orders – backlogs) drops to or below the reorder point s. This replenishment order brings the inventory position back to the order-up-to level S. Note that S consequently denotes the maximum inventory position. We define an auxiliary variable $Q = S – s$ (the actual order size is a random variable that is obviously at least equal to Q).

In general, the objective of our optimization problem is to find those values of s and S (or, equivalently, s and Q) that minimize a given cost function subject to a number of constraints:

$$\min y_0(X) \qquad (8)$$

such that

$$y_j(X) \geq a_j \ (j = 1, \ldots, m)$$
$$x_1 \geq 0, x_2 \geq 0$$

where the vector $X = (x_1, x_2)$ refers to an (s, S) (or, equivalently (s, Q)) combination.

In our two experiments we studied different objective functions and different constraints. In the next two subsections, we describe these two experiments and their results.

3.1 Experiment 1: An (s, S) Inventory System With Holding Cost And Shortage Cost Constraints

In the first experiment, the objective is to minimize the expected "total" cost $y_0(X)$ (defined as the sum of holding, shortage and ordering cost) subject to limitations on the expected holding cost $y_1(X)$ (as there is a scarcity of storage space for the product) and shortage cost $y_2(X)$ (as management wishes to maintain customer satisfaction). More specifically, we have:

$$\min y0(X) \qquad (9)$$

such that

$$y_1(X) \leq 25$$
$$y_2(X) \leq 10$$
$$x_1 \geq 0, x_2 \geq 0$$

where x_1 stands for s, and x_2 stands for S. The parameters for this (s, S) inventory control model were as follows:

- The simulated period was 120 days, with costs maintained on a \$/day basis.
- The ordering cost was \$32 per order plus \$3 per unit ordered.
- The order lead-time was uniformly distributed between 0.5 and 1: U(0.5, 1).
- The inventory review interval was one day.
- Demand was Poisson distributed with a rate of $\lambda = 10$ customers/day.
- The number of units demanded per customer varied between 1 and 4 and followed the (cumulative) distribution (0.17, 1; 0.5, 2; 0.83, 3; 1, 4).
- The holding cost was \$1/unit/day.
- The shortage cost was \$5/unit/day.

An Arena-based (s, S) model with these parameters was simulated for $r = 5$ replications at each design point in the 20-point LHD shown earlier in Figure 2. For each of these points, five replicates were simulated; each replicate started with an inventory of S units. The average results for holding, shortage and total cost are displayed in Table 1; the three rightmost columns show whether the design point violates the holding cost constraint (CVh), the shortage cost constraint (CVs), or the Boolean OR for these results (CV). The last columns shows that one-half of the 20 design points turned out to be feasible, which indicates that the design points were well placed in the experimental region.

Table 1: Simulation results for the LHD applied to the (s, S) inventory system.

Trial	s	S	Hold	Short	Total	CVh	CVs	CV
1	20	48	11.60	18.84	124.99	0	1	1
2	21	58	16.53	13.33	119.63	0	1	1
3	22	68	21.45	10.49	121.41	0	1	1
4	23	78	26.69	8.88	123.27	1	0	1
5	24	40	9.73	15.04	128.18	0	1	1
6	25	50	14.18	12.23	122.06	0	1	1
7	26	60	18.89	8.68	118.24	0	0	0
8	27	70	25.03	6.27	120.27	1	0	1
9	28	56	17.71	8.01	120.62	0	0	0
10	29	42	12.16	8.95	125.62	0	0	0
11	30	52	16.98	7.46	121.96	0	0	0
12	31	62	20.79	6.06	120.23	0	0	0
13	32	72	27.38	4.03	124.79	1	0	1
14	33	66	24.08	4.18	119.68	0	0	0
15	34	44	14.16	6.68	126.22	0	0	0
16	35	54	20.41	3.86	125.45	0	0	0
17	36	64	24.63	3.55	128.18	0	0	0
18	37	74	30.60	1.98	125.95	1	0	1
19	38	76	31.70	2.35	126.97	1	0	1
20	39	46	15.95	4.91	127.47	0	0	0

The graphical function of Minitab (2007) gave the response surfaces shown in Figure 3. The *total cost* function $y_0(X)$ is very well behaved and follows a nearly quadratic form (Minitab uses all design points to fit the surface – hence, we observe a slight departure from a perfect quad-

ratic form). The surfaces for the constraint functions for *holding cost* and *shortage cost* are highly planar in the region of interest defined by $10 \leq x_1 \leq 40$ and $40 \leq x_2 \leq 90$.

Excel *Solver* gave as the optimal solution: $x_1 = 25$, $x_2 = 63$, $y_0 = \$118.47$, $y_1 = \$20.56$, and $y_2 = \$8.13$. Hence, a response surface approach to optimization yielded an unconstrained optimum; i.e., the two constraints in (8) are not binding.

Next, we applied Lophaven et al. (2002)'s *DACE*, a Matlab Kriging toolbox, to the data x_1, x_2, and y_0; see the variables *s*, *S*, and Total in Table 1.

Figure 4 shows the plot for the total cost produced through a 20-by-20 grid of Kriging predictions. The resulting optimum is $s = 27$, $S = 60$, $y_0 = \$118.56$, $y = \$19.30$, and $y_2 = \$7.94$. Again, neither of the constraints is binding. However, this grid used only the even values of *S* over the range from 40 to 80. To find a possibly better solution, we conducted a localized Kriging search for *s* is 26, 27, and 28 and *S* is 59, 60, and 61. These Kriging predictions confirmed the above optimum.

Figure 4. Plot of Kriging predictions for the (*s*, *S*) inventory system

(a) Holding cost plot

(b) Shortage cost plot

c) Total cost plot

Figure 3. Plots for (a) Holding cost, (b) Shortage cost, and (c) Total cost for the (*s*, *S*) inventory system. Resulting from a 20-point LHD.

3.2 Experiment 2: An Integrated Production-Inventory System with an (*S*, *S*) Policy and A Service Level Constraint

The second experiment illustrates the application of Kriging to determine the optimal values of *s* and *S* in an integrated production-inventory system. In this system, every time a replenishment order is triggered, the production system needs to first produce the order before it ends up in inventory.

The optimal policy is defined as the policy that minimizes expected partial cost per time unit (consisting of inventory holding cost and ordering cost), subject to the constraint of a target Customer Service Level (CSL) (defined as the percentage of units ordered that could be immediately delivered from stock).

The production time per product unit at the factory is random. Hence, the replenishment lead-time will depend on the size of the replenishment order. Moreover, the dynamic behavior of the production system implies that replenishment orders may need to queue until the production system is available. Obviously, this influences the replenishment lead times and, consequently, the CSL that is obtained.

The model was coded entirely in the Matlab language (The MathWorks 2001). Our steady-state experiment has the following parameters:

- The simulation run length consists of 100,000 customer orders.
- Each simulation run starts with a finished goods inventory of *S* units.
- Customer orders arrive at the inventory subsystem according to a Poisson process with parameter $\lambda = 2$.
- Customer order sizes follow the shifted Poisson distribution $1 + \text{Poiss}(\mu)$ with $\mu = 2$.
- Production time for one unit at the production site is gamma distributed with mean *mX* and squared coefficient of variation *SCVX* defined as follows. *SCVX* is an input parameter, namely *SCVX* = 2. Furthermore, *mX* is based on a given target utilization rate ρ of the factory, namely $\rho = 0.8$. This ρ refers to the fraction of time that the factory will be busy, and is given by $\rho = \lambda(1 + \mu)mX$.
- The inventory holding cost per inventory unit per time unit is $c_{\text{hold}} = 0.1$.
- The cost per replenishment order is $c_{\text{order}} = 6.5$.

We studied this setting for the experimental area defined by $45 \le s \le 65$ and $40 \le Q \le 60$ where $Q = S - s$. We used Common Random Numbers (CRN) for the three random inputs (interarrival times of customer orders, customer order sizes, and unit processing times). The transient pe-riod was determined by means of Welch's method (Law and Kelton 2000, p. 520). To estimate the partial cost per time unit and the CSL for a given (*s*, *Q*) combination, we used the replication-deletion approach described in (Law and Kelton 2000, p. 525).

The initial number of replications for any input combination is set at two. Next, extra replications are added sequentially, to obtain 95 % confidence interval with a target relative error of 0.01 for the partial cost per time unit for a given (*s*, *Q*) combination. So, based on m_0 replications, we obtain

$$\bar{y} \pm t_{m_0-1;1-\frac{\alpha}{2}} \sqrt{\frac{S_y^2}{m_0}} \qquad (10)$$

with

$$\bar{y} = \frac{\sum_{j=1}^{m_0} y_j}{m_0} \qquad (11)$$

and

$$S_y^2 = \frac{\sum_{j=1}^{m_0}(y_j - \bar{y})}{m_0 - 1} \qquad (12)$$

Where

$$y_j = \frac{y_{hold,j} + y_{order,j}}{t_j} \qquad (13)$$

denotes the partial cost per time unit observed during the data collection period of the jth replication; $y_{hold,j}$ denotes the corresponding holding cost; $y_{order,j}$ denotes the ordering cost; t_j denotes the length of the data collection period in the jth replication, in time units.

For CSL we used formulas similar to (10) through (12); however, (13) is replaced by

$$CSL_j = \frac{d_j}{D_j}, \qquad (14)$$

which denotes the customer service level observed during the data collection period of the jth replication; d_j is the number of units delivered from stock, while D_j is the total number of units demanded during that same period.

Our approach in this experiment may be summarized as follows:

Step 1: *simulate* the 25 design points defined by the full factorial (or grid) with $45 \le s \le 65$ and step size 5, and $40 \le Q \le 60$ and step size 5.

Step 2: fit *Kriging* metamodels to the data from Step 1, and use these Kriging results to estimate the optimum.

Figure 5. Averaged simulated CSL for the design points in experiment 2

Figure 6. Averaged simulated partial cost per time unit for the design points in experiment 2

Step 3: *confirm* the optimum estimated in Step 2 by simulating each integer (s, Q) combination in the neighborhood of that optimum, and checking whether the Kriging model succeeds in approximating the true optimum.

Now, we discuss these steps, in more detail.

Step 1: simulate design points

Figures 5 and 6 show the resulting plots for CSL, and the objective function. Figure 5 shows that the CSL is increasing in s: higher s offers more protection against stockouts. The CSL is decreasing in Q: in our replenishment system, the replenishment lead time of an order depends on the size of the order, so larger values of Q lead to longer replenishment lead times, having an adverse effect on CSL.

Figure 6 shows that the partial cost per time unit is increasing in s, and decreasing in Q. This is the result of the combined behavior of the order cost per time unit and

Figure 7. Kriging predictors for partial cost and CSL

the holding cost per time unit. The order cost per time unit is decreasing in Q, and insensitive to s. The holding cost per time unit is increasing in s, and slightly decreasing in Q. The latter is as expected, given that large Q values in our setting lead to large replenishment batches and hence to long replenishment lead times. Consequently, with large Q values, it takes longer for the inventory to be replenished implying that in the meantime lower inventory levels are reached.

Step 2: Kriging

Next, we fit two Kriging models to the outputs observed in Step 1, namely the partial cost per time unit and the CSL (the Matlab DACE toolbox enables only univariate fitting). Figure 7 shows the Kriging predictions for the partial cost per time unit, and CSL..

The optimum based on Kriging was determined by first selecting only those integer (s, Q) combinations for which the predicted CSL is not lower than 95%; among these combinations, the one with minimum predicted partial cost was considered to be the optimum. This gives

(s^*, Q^*) = (55, 48) so S^* = 55 + 48 = 103, yielding a predicted partial cost y_0 of 5.6472 and a CSL of 0.9511.

The DACE toolbox can also be used to estimate the *gradients* of the goal (or objective) function (partial cost in this case) and the constraint (CSL) at the predicted optimum (s^* = 55, Q^* = 48); see Lophaven et al. (2002, pp. 15-16). This yields the gradients displayed in Table 2.

Table 2. Gradients of the partial cost function and the CSL function, determined through DACE

	partial cost	CSL
$\nabla f = (\frac{\partial f}{\partial s}, \frac{\partial f}{\partial Q})$	(0.0930, -0.0616)	(0.0040, -0.0027)
$\lVert \nabla f \rVert$	0.1116	0.0048
$\frac{\nabla f}{\lVert \nabla f \rVert}$	(0.8338, -0.5520)	(0.8313, -0.5558)

Dividing both gradients by their norm in Table 2 shows that the gradients point roughly in the same direction—as the Karush-Kuhn-Tucker (KKT) first-order optimality conditions require. In future research, we shall apply a statistical procedure to test whether the KKT conditions for constrained optimization indeed hold in the predicted optimum; see Bettonvil et al. (2007).

Step 3: confirmatory simulation
Finally, we compare the optima predicted by the Kriging model and an exhaustive simulation search. We therefore simulated each integer (s, Q) combination in the neighborhood of the predicted optimum; i.e., we chose to explore the experimental area defined by $54 \leq s \leq 57$ and $46 \leq Q \leq 51$, and a step size of 1 for both s and Q. This experiment gives as the optimum (s^*, Q^*) = (56, 50) with a simulated partial cost y_0 of 5.6218 and a simulated CSL of 0.95.

The minimum partial cost predicted by Kriging is close to the minimum simulated. The location of the optimum, however, does not coincide. Further analysis reveals that at (s^*, Q^*) = (56, 50) Kriging predicts a partial cost y_0 of 5.6191, which is very close to the simulated partial cost; however, the Kriging model for CSL predicts a CSL of 0.9495 so the Kriging model considers this point to be infeasible.

We believe that the latter result indicates that the precision of the Kriging methodology for constrained optimization may benefit substantially from efforts to further improve the Kriging prediction for the constraint in the neighborhood of the constraint's cut-off value. This issue will be further explored in future research.

4   CONCLUSIONS

This paper considered the problem of constrained optimization in stochastic simulation. It described two experiments in which Kriging was applied to solve this problem. Until now, Kriging has been applied chiefly to deterministic simulations. Van Beers and Kleijnen (2003) have demonstrated its application to sensitivity analysis of stochastic simulations with a single response. The present paper demonstrates that Kriging has potential for constrained optimization in stochastic simulation— though a number of issues remain to be investigated.

Firstly, future research might aim at how to improve the precision of the Kriging model for the constrained response, in the neighborhood where the constraint is binding; i.e., are the slacks significantly positive or negative? Secondly, additional work is underway to apply methods developed in Mathematical Programming to the Kriging approximations.

Simulation optimization remains a problem solved by heuristics; i.e., it is impossible to identify a truly optimum solution in stochastic simulation. For example, in multiple comparison and ranking procedures the classical assumption is that the simulation outputs are normally distributed with constant variance; hence, the procedures can only make probability statements about the optimum. In constrained simulation optimization, the problem is even more complicated. When dealing with an estimated optimum solution at a boundary, we are faced with a more complicated probability statement involving possible violations of the constraints. In current research we are using a t statistic to test the feasibility of a candidate solution, and a bootstrapped statistic to test whether the KKT conditions hold at that solution.

Unlike many other simulation optimization heuristics (which have been discussed at recent Winter Simulation Conferences), our heuristic assumes neither many simulated factor combinations nor many replications (per combination). An advantage is that our heuristic is appropriate for computationally expensive simulation experiments. A disadvantage is that we have not yet succeeded in proving the convergence of our heuristic to the true optimum.

REFERENCES

Bettonvil, B., E. del Castillo and J. P. C. Kleijnen. 2007. Statistical testing of optimality conditions in multiresponse simulation-based optimization. Working Paper, Tilburg University, Tilburg, Netherlands

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

den Hertog, D., J. P. C. Kleijnen, and A.Y.D. Siem. 2006. The correct Kriging variance estimated by bootstrapping, *Journal of the Operational Research Society*, 5(4):400-409.

Kelton, W. D., R. P. Sadowski, and D. T. Sturrock. 2007. *Simulation with Arena, Fourth Edition,* McGraw-Hill, New York.

Kleijnen, J. P. C. 2007. *DASE: Design and analysis of simulation experiments.* Springer, New York.

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,* 55(9):876-883.

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

Lophaven, S. N., H.B. Nielsen, and J. Sondergaard. 2002. Dace: A Matlab Kriging Toolbox, Technical Report IMM-TR-2002-12.

Matheron, G. 1962. Traité de géostatistique appliquée. *Memoires du Bureau de Recherches Geologiques et Minieres*, Editions Technip, Paris, 14:57-59.

McKay, M. D., R. J. Beckman, and W. J. Conover. 1979. A comparison of three methods for selecting input values in the analysis of output from a computer code. *Technometrics*, 21(2):239-245.

Minitab, Inc. 2007. *Minitab 15 statistical software,* State College, PA.

The MathWorks, Inc. 2001. *Matlab: the language of technical computing,* Novi, Michigan.

Van Beers, W. C. M. and J. P. C. Kleijnen. 2003. Kriging for interpolation in random simulation, *Journal of the Operational Research Society,* 54: 255-262.

Wackernagel, H. 2003. *Multivariate geostatistics*. Springer-Verlag, Berlin.

AUTHOR BIOGRAPHIES

WILLIAM E. BILES is the Clark Chair of Computer Aided Engineering in the Department of Industrial Engineering of the University of Louisville. He spent two years in the US Army and seven years in industrial R&D before undertaking an academic career. He has held academic appointments at the University of Notre Dame, Penn State, Louisiana State University, and the University of Louisville. Dr. Biles received the Distinguished Engineering Alumnus award from the University of Alabama in Huntsville in 2004. His most recent publication is "Environmentally Benign Manufacturing," Chapter 1 in the 2007 John Wiley handbook *Environmentally Conscious Manufacturing*. His email address is <webile01@gwise.louisville.edu>.

JACK P.C. KLEIJNEN is professor of "simulation and information systems" at Tilburg University. He is a member of the "Department of Information Management" and the "Operations Research Group" of the "Center for Economic Research (CentER)". He also teaches at the Eindhoven University of Technology, in the Postgraduate International Program in Logistics Management Systems. He is an "external fellow" of the Mansholt Graduate School of Social Sciences of Wageningen University. His research concerns the statistical design and analysis of simulation experiments, information systems, and supply chains. He has been a consultant for several organizations in the USA and Europe, and serves on any international editorial boards and scientific committees. He spent several years in the USA, at universities and private companies. He received a number of international awards, including the INFORMS Simulation Society's "Lifetime Professional Achievement Award (LPAA)" of 2005. More information can be found on his website at <http://center.uvt.nl/staff/kleijnen/>.

WIM VAN BEERS is a researcher in the Department of Information Management of Tilburg University. His primary research area is Kriging for interpolation in random simulation .He is also a member of the Department of Mathematics and Computer Science of Eindhoven University of Technology. His e-mail address is <wvbeers@uvt.nl>.

INNEKE VAN NIEUWENHUYSE is an Assistant Professor at the Faculty of Economics and Applied Economics, Department of Decision Sciences and Information Management at K.U.Leuven (Belgium). Her research focuses on the performance analysis of production and supply chain systems, using queuing theory and simulation. Her e-mail address is <inneke.vannieuwenhuyse@econ.kuleuven.be>.