Tilburg University

Design and Analysis of simulation experiments

Kleijnen, J.P.C.

Publication date:
2017

Document Version
Early version, also known as pre-print

[Link to publication in Tilburg University Research Portal](#)

Citation for published version (APA):
Kleijnen, J. P. C. (2017). *Design and Analysis of simulation experiments: Tutorial*. (CentER Discussion Paper; Vol. 2017-018). CentER, Center for Economic Research.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

No. 2017-018

**DESIGN AND ANALYSIS OF SIMULATION EXPERIMENTS:
TUTORIAL**

By

Jack P.C. Kleijnen

21 March 2017

ISSN 0924-7815
ISSN 2213-9532

Design and analysis of simulation experiments: Tutorial

Jack P.C. Kleijnen
CentER, Tilburg University, Netherlands;
<https://sites.google.com/site/kleijnenjackpc/>

March 21, 2017

Abstract

This tutorial reviews the design and analysis of simulation experiments. These experiments may have various goals: validation, prediction, sensitivity analysis, optimization (possibly robust), and risk or uncertainty analysis. These goals may be realized through metamodels. Two types of metamodels are the focus of this tutorial: (i) low-order polynomial regression, and (ii) Kriging or Gaussian processes). The type of metamodel guides the design of the experiment; this design fixes the input combinations of the simulation model. However, before a regression or Kriging metamodel is applied, the many inputs of the underlying realistic simulation model should be screened; the tutorial focuses on sequential bifurcation. Optimization of the simulated system may use either a sequence of low-order polynomials—known as response surface methodology—or Kriging models fitted through sequential designs. Finally, "robust" optimization should account for uncertainty in simulation inputs. The tutorial includes references to earlier WSC papers.

Key words: regression, Kriging, Gaussian process, factor screening, optimization, risk analysis

JEL: C0, C1, C9, C15, C44

Acknowledgement 1 *I thank the editors for inviting me to contribute a chapter to "Advances in Modeling and Simulation: Seminal Research from 50 Years of Winter Simulation Conferences", edited by A. Tolk, J. Fowler, G. Shao, and E. Yücesan.*

1 Introduction

Wagner (1975)'s famous textbook on *operations research / management science* (OR/MS) resorted to simulation "when all else fails" (p. 903) as the "method of last resort" (p. 907); nowadays, however, simulation is treated as a serious scientific method. Unfortunately, practitioners often treat their experiments with simulation models (or "simulators" or "computer codes") very unprofessionally.

Nevertheless, the simulation community is making progress; see Uhrmacher et al. (2016)'s *Winter Simulation Conference* (WSC)'s panel on the need for reproducible experiments.

In this tutorial we focus on the statistical *design and analysis of simulation experiments* (DASE). We further focus on *random* (or stochastic) simulation, because the WSCs focus on this type of simulation; nevertheless, we shall occasionally discuss deterministic simulation. By definition, random simulation gives random output (response); examples are *discrete-event dynamic systems*, *agent-based simulation*, and stochastic differential equations (see Govindan 2016). Mathematically, these simulators give random outputs because they use *pseudorandom numbers* (PRNs) as an input.

We assume that a specific simulation model is given, albeit that this model may be changed before results of the DASE are implemented. To support the managers' decisions, the simulation analysts try different values for the inputs and the parameters of this simulation model. We claim that such numerical (computer) experiments may have one or more of the following *goals*: (i) verification and validation (V & V) of the simulation model; (ii) prediction of the response of the simulated system; (iii) sensitivity analysis (SA)—either global or local—of the simulation model; (iv) optimization of the simulated real system (SimOpt); (v) risk analysis (RA) of the simulated real system. We now discuss these goals (more details including references are given in Kleijnen (2015, p. 9); in this tutorial we introduce goal ii).

Sub (i): V & V may use many methods; see Sargent et al. (2016). We focus on the use of metamodels for V & V: do the estimated effects of the metamodel have signs that agree with prior expert-knowledge about the corresponding real system; two case studies are detailed in Kleijnen (1995) and Van Ham et al. (1992). In *calibration* we estimate the combination of simulation parameters that gives the simulation output that is closest to the observed real-world output, so calibration uses optimization; see Liu et al. (2017).

Sub (ii): Prediction is necessary if experimentation with the real system is too expensive, dangerous, etc. Obviously, we can use metamodels to predict the output for an input combination that has not yet been simulated. Such prediction may be necessary if simulation takes too long; an example is real-time, on-line decision-making.

Sub (iii): SA may be performed through "experimentation" with the simulation model; i.e., simulate different "values" or "levels" for the model's parameters, input values, and starting values of the inputs (we use the terminology in Zeigler et al. (2000); i.e., we must infer the value of a parameter, whereas we can directly observe the value of an input). For example, in a queueing simulation we may start with an "empty" (no waiting customers) simulated system, exponential interarrival times with a fixed arrival rate, and a fixed number of servers. Altogether we speak of *factors*, which have fixed values during one "run" of the simulation experiment.

When we experiment, we use a *metamodel*; i.e., an explicit and relatively simple approximation of the *input / output* (I/O) function implicitly defined by the underlying simulation model. We treat the simulation model as a *black*

box; i.e., we observe only the I/O of the simulation model. Actually, simulation analysts may not realize that they are using a metamodel. For example, if we change only one factor at a time, then we assume that the factors do not interact. When we consider only two levels for each of our (say) k factors, then a 2^k design implies that we assume a metamodel with interactions among these k factors—but without "pure" higher order effects such as purely quadratic or cubic effects. In this tutorial we shall discuss metamodels in much detail.

Sub (iv) and (v): We may apply RA for (a) a single combination of the simulation inputs, (b) a given set of combinations, (c) the best combination estimated through SimOpt. Alternative (b) is described in Kleijnen (2015, p. 221) using a first-order polynomial metamodel to estimate which combinations of uncertain inputs form the frontier that separates acceptable and unacceptable outputs. Alternative (c) is detailed in Kleijnen (2015, p. 275). However, we may also try to estimate the optimum while accounting for uncertainty in the parameters of the simulation; we call this approach *robust optimization* (RO). We distinguish two RO approaches; namely, the Taguchi approach and the mathematical programming (MP) approach. Taguchi has been popular in mechanical engineering, since several decades. In MP, RO is a recent important topic. We expect that both approaches will inspire OR/MS. For example, Taguchi inspired Dellino et al. (2009) and Dellino et al. (2010). An example of recent RO in MP is Bertsimas and Misić (2017). A recent example of RO combining Taguchi and MP is Yanikoglu et al. (2017). Approaches related to RO are also discussed in Nelson and Song (2017) and Zhou (2017).

The *goals of metamodels* immediately follow from the five goals of the underlying simulation models. Because Kleijnen (2015, p.9) does not discuss goal 2, we point out that a simulation may be computationally *slow* or *expensive*: i.e., it may take much computer time to obtain the response for a given combination of the simulation inputs. An example is the deterministic simulator of a car-crash model at Ford that required 36 to 160 hours of computer time; see Simpson et al. (2004). Another example is the discrete-event simulation of a *rare* event with dramatic consequences (e.g., a nuclear accident); without adequate importance sampling the simulation needs very many replications. Obviously, a metamodel can quickly predict the output for a new input combination of the expensive simulation model.

There are important differences between "traditional" *design of experiments* (DOE) and DASE. Kleijnen (2015, p. 12) mentions that DOE was developed for real-world (nonsimulated) experiments in agriculture, engineering, psychology, etc. In these experiments it is impractical to investigate "many" factors; i.e., ten factors seems a maximum. Moreover, it is hard to investigate factors with more than "a few" levels; five levels per factor seems the limit. In simulation, however, these restrictions do not apply; i.e., simulation models may have thousands of factors—each with many values. Consequently, a multitude of factor combinations may be simulated. Moreover, simulation is well-suited to "sequential" designs instead of "one shot" designs, because simulation experiments run on computers that typically produce output sequentially (apart from parallel computers; see Gramacy (2015)), whereas agricultural experiments run during

a single growing season. Altogether, many simulation analysts need a change of mindset.

Mathematically, we consider $w = f_{\text{sim}}(\mathbf{z}, \mathbf{r})$ where w denotes the simulation output (e.g., average waiting time), f_{sim} the I/O function of the simulation model, $\mathbf{z}$ the vector with the values of the k simulation inputs, and $\mathbf{r}$ the vector with PRNs used by the random simulation model; in deterministic simulation $\mathbf{r}$ vanishes. Usually, $\mathbf{z}$ is *standardized* (scaled) such that the resulting input $\mathbf{d}$ has elements either $-1 \leq d_j \leq 1$ or $0 \leq d_j \leq 1$ with $j = 1, \dots, k$.

Note: An input of a queuing simulation may be the traffic rate—not the arrival rate $1/\mu_a$ and the service rate $1/\mu_s$ where μ denotes the mean (expected value)—so $z_1 = \mu_s/\mu_a$. Another input may be the priority rule (queueing discipline), which is a qualitative input with at least two levels; e.g., first-in-first-out and smallest-service-time-first. A qualitative input with more than two values needs special care; see Kleijnen (2015, pp. 69–71).

The function $w = f_{\text{sim}}(\mathbf{z}, \mathbf{r})$ may be approximated by $y = f_{\text{meta}}(\mathbf{x}) + e$ where y is the metamodel output, which may deviate from w so the approximation error e may have $\mu_e \neq 0$. If $\mu_e = 0$, then we call f_{meta} *adequate* (or valid). There are many types of metamodels; see Kleijnen (2015, p. 10). We, however, focus on two types: *low-order polynomials*, which are linear regression models, and *Kriging* or Gaussian process (GP) models.

Note: Mitchell and Morris (1992) introduced Kriging into the WSC community. Later on, Van Beers and Kleijnen (2004) surveyed Kriging for simulation. Biles et al. (2007) used Kriging in constrained simulation optimization. Kleijnen et al. (2012) introduced convex and monotonic bootstrapped Kriging.

We base this tutorial on Kleijnen (2017), which is a recent review based on Kleijnen (2015). We update that review; e.g., we add Binois et al. (2017) in Section 5. Furthermore, we emphasize publications in the WSC proceedings. Kleijnen (2015) includes hundreds of additional references, and many website addresses for software.

We organize this chapter as follows. Section 2 summarizes classic linear regression and DOE. Section 3 presents solutions for DASE if the classic statistical assumptions are violated in practice. Section 4 summarizes *sequential bifurcation* (SB) for the screening of a multitude of factors. Section 5 summarizes Kriging and its designs. Section 6 summarizes SimOpt using either low-order polynomials or Kriging, including RO.

2 Basic linear regression and designs

Because we assume that the readers are familiar with the basics of linear regression, we limit our discussion of this regression to definitions of our mathematical symbols and terminology. We define the *linear regression* (meta)model $\mathbf{y} = \mathbf{X}_N \boldsymbol{\beta} + \mathbf{e}$ where $\mathbf{y}$ denotes the N -dimensional vector with the dependent (explained) variable (also see Section 1), N denotes the number of observations on the simulation response with $N = \sum_{i=1}^n m_i$, n the number of simulated input combinations, m_i the number of replications for combination i (obviously,

deterministic simulation implies $m_i = 1$ so $N = n$), $\mathbf{X}_N$ is the $N \times q$ matrix of independent (explanatory) regression variables—obviously, $\mathbf{X}_N$ has m_i identical rows, whereas $\mathbf{X}_n$ denotes the corresponding $n \times q$ matrix without any identical rows determined by the $n \times k$ design matrix $\mathbf{D}$ and the type of regression model (e.g., second-order polynomial), $\boldsymbol{\beta}$ denotes the q -dimensional vector with regression parameters (coefficients), and $\mathbf{e}$ denotes the N -dimensional vector with the residuals $\mathbf{E}(\mathbf{y}) - \mathbf{E}(\mathbf{w})$ where $\mathbf{w}$ denotes the N -dimensional vector with independent simulation outputs; this independence requires that random simulation does not use *common random numbers* (CRN). We define the q -dimensional row vector $\mathbf{x}_i = (x_{i;1}, \dots, x_{i;q})$. Obviously, n denotes the number of *different* combinations of the k simulation inputs. We focus on a special type of linear regression; namely, a *second-order polynomial* with k simulation inputs, which has an intercept β_0 , k first-order effects β_j ($j = 1, \dots, k$), $k(k-1)/2$ two-factor interactions (cross-products) $\beta_{j;j'}$ ($j' > j$), and k purely quadratic effects $\beta_{j;j}$. These *interactions* mean that the effect of an input depends on the values of one or more other inputs. A *purely quadratic* effect means that the marginal effect of the input is not constant, but diminishes or increases. This polynomial is nonlinear in $\mathbf{x}$, and linear in $\boldsymbol{\beta}$. We assume that interactions among three or more inputs are unimportant, because such interactions are hard to interpret and in practice are indeed often unimportant. Of course, we should check this assumption; i.e., we should "validate" the estimated metamodel (see the next section).

The *ordinary least squares* (OLS) estimator of $\boldsymbol{\beta}$ is $\hat{\boldsymbol{\beta}} = (\mathbf{X}_N' \mathbf{X}_N)^{-1} \mathbf{X}_N' \mathbf{w}$, assuming the inverse of $\mathbf{X}_N' \mathbf{X}_N$ exists; e.g., $\hat{\boldsymbol{\beta}}$ exists if $\mathbf{X}_N$ is *orthogonal*. If m_i is a positive integer constant (say) m , then we may replace $\mathbf{w}$ by $\bar{\mathbf{w}}$ with the n elements $\bar{w}_i = \sum_{r=1}^m w_{i;r}/m$ and replace $\mathbf{X}_N$ by $\mathbf{X}_n$. Moreover, $\hat{\boldsymbol{\beta}}$ is the *maximum likelihood estimator* (MLE) if $\mathbf{e}$ is *white noise*, which means that $\mathbf{e}$ is *normally, independently, and identically distributed* (NIID) with zero mean and constant variance σ_e^2 ; i.e., $\mathbf{e} \sim N_N(\mathbf{0}_N, \sigma_e^2 \mathbf{I}_{N \times N})$ where N_N stands for N -variate (see subscript) normally (symbol: N) distributed, $\mathbf{0}_N$ for the N -variate vector with zeroes, and $\mathbf{I}_{N \times N}$ for the $N \times N$ identity matrix. If the metamodel is valid, then $\sigma_e^2 = \sigma_w^2$. White noise implies that $\hat{\boldsymbol{\beta}}$ has the $q \times q$ covariance matrix $\boldsymbol{\Sigma}_{\hat{\boldsymbol{\beta}}} = (\mathbf{X}_N' \mathbf{X}_N)^{-1} \sigma_w^2$. Because σ_w^2 is unknown, we estimate $\sigma_w^2 = \sigma_e^2$ through the *mean squared residuals* (MSR) $(\hat{\mathbf{y}} - \mathbf{w})'(\hat{\mathbf{y}} - \mathbf{w})/(N - q)$ with predictor $\hat{\mathbf{y}} = \mathbf{X}_N \hat{\boldsymbol{\beta}}$ and degrees of freedom (DOF) $N - q > 0$; this inequality is satisfied, even if $n = q$ but $m_i > 1$ for at least one value of i . This MSR gives the estimator $\hat{\boldsymbol{\Sigma}}_{\hat{\boldsymbol{\beta}}}$. This $\hat{\boldsymbol{\Sigma}}_{\hat{\boldsymbol{\beta}}}$ has a main diagonal with the elements $s^2(\hat{\beta}_g)$ ($g = 1, \dots, q$), which give the square roots $s(\hat{\beta}_g)$. *Confidence intervals* (CIs) and *tests* for the individual $\hat{\beta}_g$ follow from the Student t -statistic with $N - q$ DOF: $t_{N-q} = (\hat{\beta}_g - \beta_g)/s(\hat{\beta}_g)$. Finally, $\hat{\mathbf{y}} = \mathbf{X}_N \hat{\boldsymbol{\beta}}$ implies $s^2(\hat{y}|\mathbf{x}_i) = \mathbf{x}_i' \hat{\boldsymbol{\Sigma}}_{\hat{\boldsymbol{\beta}}} \mathbf{x}_i$. This $s^2(\hat{y}|\mathbf{x}_i)$ is minimal at the center of the experimental area. So, if the goal of the metamodel is to predict the simulation output at a specific "new" point, then we should simulate N "old" points close to that new point (a similar guideline applies for Kriging; see Gramacy (2015)).

Because $\Sigma_{\hat{\beta}} = (\mathbf{X}'_N \mathbf{X}_N)^{-1} \sigma_w^2$ we may select $\mathbf{X}_N$ such that we "optimize" $\Sigma_{\hat{\beta}}$. Obviously, $\mathbf{X}_N$ is determined by $\mathbf{X}_n$, m_i , and the type of regression model (e.g., $\mathbf{X}_N$ may include x_j^2). DOE does not say much about the selection of m_i ; typically, DOE assumes $m_i = 1$. If $m_i = m \geq 1$, then an orthogonal $\mathbf{X}_n$ implies an orthogonal $\mathbf{X}_N$ (we may specify $\mathbf{X}_N$ as $\mathbf{X}_n$ "stapled" or "stacked" m times). We shall further discuss m_i in Section 3.3. In the next subsections we shall discuss the following special types of second-order polynomial regression-model: (i) $\beta_{j;j'} = 0$ with $j \leq j'$; (ii) $\beta_{j;j} = 0$; we discuss (a) $\hat{\beta}_j$ unbiased by $\beta_{j;j'}$ with $j < j'$, and (b) $\hat{\beta}_j$ and $\hat{\beta}_{j;j'}$ unbiased. These types of polynomials require designs of different *resolution* (abbreviated to R). Obviously, $\mathbf{X}_n$ is determined by $\mathbf{D}$. To select a specific $\mathbf{D}$ with $\mathbf{z}$ constrained over a given experimental area, we try to minimize $\text{Var}(\hat{\beta}_g)$ ($g = 1, \dots, q$); other criteria are discussed in Kleijnen (2015, pp. 66–67). We can prove that this minimization requires an orthogonal $\mathbf{X}_N$, which gives $\Sigma_{\hat{\beta}} = (N\mathbf{I})^{-1} \sigma_w^2 = \mathbf{I} \sigma_w^2 / N$. This $\Sigma_{\hat{\beta}}$ implies that the $\hat{\beta}_j$ are statistically independent. Moreover, these $\hat{\beta}_j$ have the same variance (σ_w^2 / N). So we can *rank* the explanatory variables in order of importance, using either $\hat{\beta}_g$ or t_{N-q} with $\beta_g = 0$ (so $t_{N-q} = \hat{\beta}_g / s(\hat{\beta}_g)$); obviously, we do not hypothesize that the intercept β_0 is zero. Usually, DOE assumes that $z_{i;j}$ is standardized such that $-1 \leq d_{i;j} \leq 1$.

2.1 R-III designs for first-order polynomials

If a first-order polynomial is an adequate metamodel (so $\beta_{j;j'} = 0$ with $j \leq j'$), then *changing one factor at a time* gives unbiased $\hat{\beta}_j$ ($j = 1, \dots, k$). However, these $\hat{\beta}_j$ do not have minimum variances; a R-III design does minimize these variances because this design gives an orthogonal $\mathbf{X}_N$, as we shall see in this subsection. Furthermore, we can prove that $\text{Var}(\hat{\beta}_j)$ is minimal if we simulate only two levels per factor, as far apart as the experimental area allows (in either a one-factor-at-a-time or a R-III design); see Kleijnen (2015, pp. 44–49).

Note: If all $\hat{\beta}_g$ are independent, then the *full* regression model with q effects and the *reduced* model with nonsignificant effects eliminated have *identical* values for those estimated effects that occur in both models. If not all $\hat{\beta}_g$ are independent, then so-called *backwards elimination* of nonsignificant effects changes the values of the remaining estimates.

R-III designs are also known as *Plackett-Burman* (PB) designs. A subclass are *fractional factorial two-level* designs, denoted by 2_{III}^{k-p} with integer p such that $0 \leq p < k$ and $2^{k-p} \geq 1 + k$: we first discuss 2_{III}^{k-p} designs. Any 2^{k-p} design is *balanced*; i.e., each input is simulated $n/2$ times at its lower value L_j and at its higher value H_j . Furthermore, a 2^{k-p} design gives an *orthogonal* $\mathbf{X}_n$. A R-III design may be *saturated*: $N = q$ (with $N = \sum_{i=1}^n m_i$, $n = 2^{k-p}$, and $q = 1 + k$). A saturated design implies that the MSR is undefined (because $N - q = 0$). To solve this problem, we may obtain replications for one or more combinations of the R-III design; e.g., the combination at the *center* of the experiment where $d_j = 0$ if d_j is quantitative and d_j is randomly selected as

-1 or 1 if d_j is qualitative with two levels. A simple algorithm for constructing 2_{III}^{k-p} designs is given in Kleijnen (2015, pp. 53–54).

Obviously, 2_{III}^{k-p} designs have n equal to a power of 2. However, there are also PB designs with n a multiple of 4; e.g., $8 \leq k \leq 11$ implies $n = 12$ (whereas 2_{III}^{12-8} implies $n = 16$). If $8 \leq k < 11$, then we do not use $n - (k + 1)$ columns of $\mathbf{D}$. Actually, there are PB designs for $12 \leq n \leq 96$; for $12 \leq n \leq 36$ these designs are tabulated in Montgomery (2009, p. 326) and Myers et al. (2009, pp. 165). A PB design is balanced and gives an orthogonal $\mathbf{X}_n$.

2.2 R-IV designs for first-order polynomials

A R-IV design gives $\hat{\beta}_j$ unbiased by $\beta_{j;j'}$ with $j < j'$ (so $q = 1 + k + k(k-1)2$). Such a design uses the *foldover theorem*; i.e., augmenting a R-III design $\mathbf{D}$ with its *mirror* design $-\mathbf{D}$ gives a R-IV design (we shall also apply the foldover trick in Section 4 on screening). A R-IV design does not enable unbiased estimators of the $k(k-1)2$ *individual* two-factor interactions. For example, a 2_{III}^{7-4} design has $n_{III} = 8$ so $n_{IV} = 16$; furthermore, $q = 29$ so $n_{IV} < q$ and we cannot compute the OLS estimators of the 29 individual regression parameters.

2.3 R-V designs for two-factor interactions

A R-V design enables unbiased $\hat{\beta}_j$ and $\hat{\beta}_{j;j'}$ with $j < j'$. Obviously, we have $q = 1 + k + k(k-1)/2$. The DOE literature gives tables for generating 2_V^{k-p} designs. Unfortunately, these designs are not saturated at all; e.g., the 2_V^{8-2} design implies $n = 64 \gg q = 37$. We point out that *Rechtschaffner* designs do include saturated R-V designs; see Kleijnen (2015, pp.62–63). We shall use R-V designs in the next subsection.

2.4 CCDs for second-degree polynomials

A CCD or *central composite design* enables unbiased $\hat{\beta}_j$ and $\hat{\beta}_{j;j'}$ with $j \leq j'$. A CCD consists of three subdesigns: (i) a R-V design; (ii) the *central* combination $\mathbf{0}'_k$; (iii) the $2k$ *axial* combinations—which form a *star design*—with $d_j = c$ and $d_{j'} = 0$ where $j' \neq j$, and $d_j = -c$ and $d_{j'} = 0$. Obviously, $c \neq 1$ implies five values per input, whereas $c = 1$ implies three values per input. The usual choice of c is not 1. The "optimal" choice of c assumes white noise, which does not hold in practice so we do not detail this choice. Finally, if $c \leq 1$, then $-1 \leq d_{i;j} \leq 1$; else $-c \leq d_{i;j} \leq c$.

A CCD gives a *non-orthogonal* $\mathbf{X}_n$; e.g., any two columns corresponding with β_0 , $\beta_{j;j}$, and $\beta_{j';j'}$ are not orthogonal. A CCD is rather inefficient (i.e., $n \gg q$); yet, CCDs are popular in DOE, especially in *response surface methodology* (RSM) (see Section 6.1). For further discussion of CCDs and other types of designs for second-degree polynomials we refer to Kleijnen (2015, pp. 64–66), and Myers et al. (2009, pp. 296–317). A more efficient modified CCD is derived in Kleijnen and Shi (2017).

3 Classic assumptions vs. simulation practice

The *classic* assumptions stipulate a single type of output (univariate output) and white noise. In simulation practice, however, the simulation model often has a *multivariate* output and no white noise—as we discuss now.

3.1 Multivariate simulation output

We assume that for r -variate ($r \geq 1$) simulation output we use r univariate linear regression metamodels $\mathbf{y}^{(l)} = \mathbf{X}_N^{(l)} \beta^{(l)} + \mathbf{e}^{(l)}$ with $l = 1, \dots, r$ where $\mathbf{y}^{(l)}$ corresponds with output type l ; $\mathbf{X}_N^{(l)}$ is the $N \times q_l$ matrix for metamodel l ; $\beta^{(l)}$ is the vector with the q_l regression parameters for metamodel l ; and $\mathbf{e}^{(l)}$ is the N -dimensional vector with the residuals of metamodel l . More restrictively we assume that all r metamodels are polynomials of the same order (e.g., second-order), so $\mathbf{X}^{(l)} = \mathbf{X}$ and $q_l = q$. Obviously, $\mathbf{e}^{(l)}$ has variances that may vary with l (e.g., the variances differ for simulated inventory costs and service percentages), and $e_i^{(l)}$ and $e_i^{(l')}$ are not independent (they are different transformations of the same PRNs). Nevertheless, $\mathbf{X}^{(l)} = \mathbf{X}$ implies that the *best linear unbiased estimator* (BLUE) of $\beta^{(l)}$ is the OLS estimator computed per output: $\hat{\beta}^{(l)} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{w}^{(l)}$. Furthermore, CIs and tests for the elements in $\hat{\beta}^{(l)}$ use the classic formulas in the preceding section. We are not aware of any *general* designs for multivariate output. For further discussion of multivariate output we refer to Kleijnen (2015, pp. 85–88).

3.2 Nonnormal simulation output

The normality assumption often holds *asymptotically*; i.e., if the simulation run is long, then the sample average of the autocorrelated observations is nearly normal. Estimated quantiles, however, may be very nonnormal, especially in case of an "extreme" (e.g., 99%) quantile. The t -statistic (used in the CIs) is quite insensitive to nonnormality. Whether the actual simulation run is long enough to make the normality assumption hold, is always hard to know. Therefore it seems good practice to test whether the simulation output has a Gaussian *probability density function* (PDF). We may then use various *residual plots* and *goodness-of-fit statistics*; e.g., the chi-square statistic. We must then assume that the tested outputs are IID. We may therefore obtain "many" (say, 100) replications for a specific input combination (e.g., the base scenario). However, if the simulation is expensive, then these plots are too rough and these tests have no power.

Actually, the white-noise assumption concerns e , not w . Given $m_i \geq 1$ replications ($i = 1, \dots, n$), we obtain $\bar{w}_i = \sum_{r=1}^{m_i} w_{i,r} / m_i$ and the corresponding $\hat{e}_i = \hat{y}_i - \bar{w}_i$. For simplicity of presentation, we assume that m_i is a constant m . If $w_{i,r}$ has a constant variance σ_w^2 , then $\bar{w}_i$ also has a constant variance; namely, $\sigma_{\bar{w}}^2 = \sigma_w^2 / m$. Unfortunately, even if $\bar{w}_i$ has a constant variance $\sigma_{\bar{w}}^2$ and is independent of $\bar{w}_{i'}$ with $i \neq i'$ (no CRN), then $\Sigma_{\hat{e}} = [\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}']\sigma_{\bar{w}}^2$

so $\hat{\mathbf{e}}$ does not have IID components; so, the interpretation of the popular plot with estimated residuals is not straightforward.

We may apply *normalizing transformations*; e.g., $\log(w)$ may be more normally distributed than w . Unfortunately, the metamodel now explains the behavior of the transformed output—not the original output; also see Kleijnen (2015, p. 93).

A statistical method that does not assume normality is *distribution-free bootstrapping* or *nonparametric bootstrapping*. This bootstrapping may be used to examine (i) nonnormal distributions, or (ii) nonstandard statistics (e.g., R^2). We denote the *original observations* by w , and the *bootstrapped observations* by w^* . We assume that these w are IID; indeed, $w_{i;1}, \dots, w_{i;m_i}$ are IID because the m_i replications use nonoverlapping PRN streams. We *resample—with replacement*—these m_i observations such that the original sample size m_i remains unchanged. We apply this resampling to each combination i . The resulting $w_{i;1}^*, \dots, w_{i;m_i}^*$ give the average $\bar{w}_i^*$, which give the n -dimensional vector $\bar{\mathbf{w}}^*$. For simplicity's sake, we now assume $m_i = m > 1$, so the bootstrapped OLS estimator of β is $\hat{\beta}^* = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\bar{\mathbf{w}}^*$. To reduce sampling error, we select a *bootstrap sample size* (say) B , and repeat this resampling B times; e.g., B is 100 or 1,000. This B gives $\hat{\beta}_b^*$ with $b = 1, \dots, B$. For simplicity's sake, we focus on β_q (last element of β). To compute a two-sided $(1 - \alpha)$ CI, the *percentile method* computes the $\alpha/2$ quantile (or percentile) of the *empirical density function* (EDF) of $\hat{\beta}_q^*$ obtained through sorting the B observations on $\hat{\beta}_{q;b}^*$. This sorting gives the *order statistics*, denoted by the subscript $(\cdot)$ where—for notational simplicity—we assume that $B\alpha/2$ is integer so the estimated $\alpha/2$ quantile is $\hat{\beta}_{q;(B\alpha/2)}^*$. Analogously we obtain $\hat{\beta}_{q;(B[1-\alpha/2])}^*$. These two quantiles give a two-sided asymmetric $(1 - \alpha)$ CI: $\hat{\beta}_{q;(B\alpha/2)}^* < \beta_q < \hat{\beta}_{q;(B[1-\alpha/2])}^*$. We shall mention more bootstrap examples, in later sections.

3.3 Heterogeneous variances of simulation outputs

In practice, $\text{Var}(w_i)$ changes as $\mathbf{x}_i$ changes ($i = 1, \dots, n$). In some applications, however, we may hope that this variance heterogeneity is negligible. Unfortunately, $\text{Var}(w_i)$ is unknown so we must estimate it. The classic unbiased estimator is $s^2(w_i) = \sum_{r=1}^{m_i} (w_{i;r} - \bar{w}_i)^2 / (m_i - 1)$. This $s^2(w_i)$ itself has a high variance. To compare the n estimators $s^2(w_i)$, we can apply many tests; see Kleijnen (2015, p. 101).

If we either assume or find variance heterogeneity, then we may still use *OLS*. Actually, $\hat{\beta}$ is still *unbiased*, but $\Sigma_{\hat{\beta}}$ becomes $(\mathbf{X}'_n \mathbf{X}_n)^{-1} \mathbf{X}'_n \Sigma_{\bar{\mathbf{w}}} \mathbf{X}_n (\mathbf{X}'_n \mathbf{X}_n)^{-1}$ where for simplicity's sake we assume $m_i = m$ so $\Sigma_{\bar{\mathbf{w}}}$ is the $n \times n$ diagonal matrix with the main-diagonal elements $\text{Var}(w_i)/m$.

The DOE literature ignores designs for heterogeneous output variances. We propose classic designs with m_i such that we obtain approximately constant $s^2(w_i)/m_i$ ($i = 1, \dots, n$). Initially we take a *pilot sample* of size $m_0 \geq 2$ for each combination, which gives (say) $s_i^2(m_0)$. Next we select a number of additional

replications $\hat{m}_i - m_0$ with

$$\hat{m}_i = m_0 \times \text{nint} \left[\frac{s_i^2(m_0)}{\min_i s_i^2(m_0)} \right] \quad (1)$$

where $\text{nint}[x]$ denotes the integer closest to x . Combining the $\hat{m}_i$ replications of the two stages gives $\bar{w}_i$ and $s^2(w_i)$. This $\bar{w}_i$ gives $\hat{\beta}$, while $s^2(w_i)$ gives the diagonal matrix $\hat{\Sigma}_{\bar{\mathbf{w}}}$ with main-diagonal elements $s_i^2(\hat{m}_i)/\hat{m}_i$. This $\hat{\Sigma}_{\bar{\mathbf{w}}}$ gives $\hat{\Sigma}_{\hat{\beta}}$, which—together with t_{m_0-1} —gives a CI for β_j .

Actually, (1) gives the *relative* number of replications $\hat{m}_i/\hat{m}_{i'}$. To select absolute numbers, we recommend the rule in Law (2015, p. 505) with number of replications $\hat{m}$ and a relative estimation error r_{ee} :

$$\hat{m} = \min \left[r \geq m : \frac{t_{r-1; 1-\alpha/2} \sqrt{s_i^2(m)/r}}{|\bar{w}(m)|} \leq \frac{r_{ee}}{1 + r_{ee}} \right]. \quad (2)$$

3.4 Common random numbers

CRN are often applied; actually, CRN are the default in software for discrete-event simulation. Given $m_i = m$, we can arrange the simulation output $w_{i;r}$ with $i = 1, \dots, n$ and $r = 1, \dots, m$ into a matrix $\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_m)$ with $\mathbf{w}_r = (w_{1;r}, \dots, w_{n;r})'$. Obviously, CRN create correlation between $w_{i;r}$ and $w_{i';r}$. Moreover, different replications use nonoverlapping PRN streams so $w_{i;r}$ and $w_{i';r'}$ with $r \neq r'$ —or the n -diagonal vectors $\mathbf{w}_r$ and $\mathbf{w}_{r'}$ —are independent. CRN are used to reduce $\text{Var}(\hat{\beta}_g)$ and $\text{Var}(\hat{y})$; unfortunately, CRN increase the variance of the estimated intercept. For details on the effective usage of CRN we refer to Law (2015, pp. 592-604).

To compute $\hat{\beta}$, we do not use $\mathbf{W}$; i.e., we use the vector $\mathbf{w}$ with $N = \sum_{i=1}^n m_i$ elements. To compute $\hat{\Sigma}_{\hat{\beta}}$ in case of CRN, we use the non-diagonal matrix $\hat{\Sigma}_{\mathbf{w}}$. Unfortunately, this $\hat{\Sigma}_{\mathbf{w}}$ is *singular* if $m \leq n$; if $m > n$, then we may compute CIs for $\hat{\beta}_j$ from t_{m-1} . An alternative method requires only $m > 1$, and computes $\hat{\beta}_r = (\mathbf{X}'_n \mathbf{X}_n)^{-1} \mathbf{X}'_n \mathbf{w}_r$ ($r = 1, \dots, m$). We again focus on a single element of this $\hat{\beta}_r$; namely, element g ($g = 1, \dots, q$). Obviously, $\hat{\beta}_{g;r}$ and $\hat{\beta}_{g;r'}$ with $r \neq r'$ and $r' = 1, \dots, m$ are IID with variance $\text{Var}(\hat{\beta}_g)$. The m replications give $\bar{\hat{\beta}}_g = \sum_{r=1}^m \hat{\beta}_{g;r}/m$ and $s^2(\bar{\hat{\beta}}_g) = \sum_{r=1}^m (\hat{\beta}_{g;r} - \bar{\hat{\beta}}_g)^2/[m(m-1)]$; together they give $t_{m-1} = (\bar{\hat{\beta}}_g - \beta_g)/s(\bar{\hat{\beta}}_g)$.

Unfortunately, we cannot apply this alternative when estimating a *quantile* instead of a mean. We then recommend distribution-free bootstrapping; see Kleijnen (2015, p. 99, 110). Furthermore, m_i is not a constant if we select m_i such that $\bar{w}_i$ has the same—absolute or relative—width of the CI around $\bar{w}_i$; see again (2). We must then adjust the analysis; see Kleijnen (2015, p. 112).

3.5 Validation of metamodels

We discuss various validation methods (which we may also use to compare first-order against second-order polynomials, or linear regression against Kriging metamodels). One method uses $R^2 = \sum_{i=1}^n (\hat{y}_i - \bar{w})^2 / \sum_{i=1}^n (\bar{w}_i - \bar{w})^2 = 1 - \sum_{i=1}^n (\hat{y}_i - \bar{w}_i)^2 / \sum_{i=1}^n (\bar{w}_i - \bar{w})^2$ where $\bar{w} = \sum_{i=1}^n \bar{w}_i / n$ and $m_i \geq 1$. If $n = q$ (saturated design), then $R^2 = 1$ —even if $E(\hat{e}_i) \neq 0$. If $n > q$ and q increases, then R^2 increases—whatever the size of $|E(\hat{e}_i)|$ is; because of possible *overfitting*, we may therefore use the *adjusted* R^2 : $R_{\text{adj}}^2 = 1 - (1 - R^2)(n - 1)/(n - q)$. Unfortunately, we do not know *critical values* for R^2 or R_{adj}^2 . We might either use subjective lower thresholds, or estimate the distributions of these two statistics through distribution-free bootstrapping; see Kleijnen (2015, p. 114).

Actually, we prefer *cross-validation*. We again suppose that $m_i = m \geq 1$ so we may replace $\mathbf{w}$ by $\bar{\mathbf{w}}$ in OLS. Now we delete I/O combination i to obtain $(\mathbf{X}_{-i}, \bar{\mathbf{w}}_{-i})$ where we suppress the subscript n of $\mathbf{X}$. Next we compute $\hat{\beta}_{-i} = (\mathbf{X}_{-i}' \mathbf{X}_{-i})^{-1} \mathbf{X}_{-i}' \bar{\mathbf{w}}_{-i}$ ($i = 1, \dots, n$). This gives $\hat{y}_{-i} = \mathbf{x}_i' \hat{\beta}_{-i}$. We may "eyeball" the *scatterplot* with $(\bar{w}_i, \hat{y}_{-i})$, and decide whether the metamodel is valid. Alternatively, we may compute the *Studentized prediction error*

$$t_{m-1}^{(i)} = \frac{\bar{w}_i - \hat{y}_{-i}}{\sqrt{s^2(\bar{w}_i) + s^2(\hat{y}_{-i})}} \quad (3)$$

where $s^2(\bar{w}_i) = s^2(w_i)/m$ and $s^2(\hat{y}_{-i}) = \mathbf{x}_i' \hat{\Sigma}_{\hat{\beta}_{-i}} \mathbf{x}_i$ with $\hat{\Sigma}_{\hat{\beta}_{-i}} = s^2(\bar{w}_i)(\mathbf{X}_{-i}' \mathbf{X}_{-i})^{-1}$.

We reject the metamodel if $\max_i |t_{m-1}^{(i)}| > t_{m-1; 1-\alpha/(2n)}$ where we use the *Bonferroni inequality* so we replace $\alpha/2$ by $\alpha/(2n)$; i.e., we control the *experimentwise* or *familywise* type-I error rate α . Note that regression software uses a shortcut to avoid the n recomputations in cross-validation.

Cross-validation affects not only $\hat{y}_{-i}$, but also $\hat{\beta}_{-i}$ (see above). We may be interested not only in the predictive performance of the metamodel, but also in its *explanatory* performance; i.e., do the n estimates $\hat{\beta}_{-i}$ remain stable?

Related to cross-validation are *diagnostic* statistics; e.g., the *prediction sum of squares* (PRESS): $[\sum_{i=1}^n (\hat{y}_{-i} - w_i)^2 / n]^{1/2}$. We may apply bootstrapping to estimate the distribution of the various validation statistics; see Kleijnen (2015, p. 120).

If the validation suggests an unacceptable fitting error e , then we may consider various *transformations*. For example, we may replace y and x_j by $\log(y)$ and $\log(x_j)$ ($j = 1, \dots, k$) so that the first-order polynomial approximates relative changes through k *elasticity coefficients*. If we assume that f_{sim} is monotonic, then we may replace w and x_j by their ranks: *rank regression*. In the preceding subsections, we also considered transformations that make w better satisfy the assumptions of normality and variance homogeneity; unfortunately, different objectives of a transformation may conflict with each other.

In Section 2 we discussed designs for low-order polynomials. If such a design does not give a valid metamodel, then we do not recommend routinely adding higher-order terms: these terms are hard to interpret. However, if the goal is

better prediction, then we may add higher-order terms; e.g., a 2^k design enables the estimation of the interactions among three or more inputs. However, adding more terms may lead to overfitting; see our comment on R_{adj}^2 . A detailed example of fitting a polynomial in simulation optimization is presented in Barton (1997, Fig. 4). Adding more explanatory variables is called *step-wise regression*, whereas eliminating nonsignificant variables is called *backwards elimination*, which we briefly discussed in the Note in Section 2.1.

4 Factor screening: sequential bifurcation

Factor screening—or briefly screening—searches for the really important inputs (or factors) among the many inputs that can be varied in a (simulation) experiment. It is realistic to assume *sparsity*, which means that only a few inputs among these many inputs are really important; the *Pareto* principle or *20-80* rule states that only a few inputs—say, 20%—are really important. Two examples with 281 and 92 inputs respectively are presented in Kleijnen (2015, p. 136); screening finds only 15 and 11 inputs to be really important. In practice, we should apply screening *before* we apply the designs and analyses of the preceding two sections; however, to explain screening methods, we need those two sections.

Four types of screening designs that use different mathematical assumptions, are summarized in Kleijnen (2015, pp. 137–139). We, however, focus on SB, because SB is very efficient and effective if its assumptions are satisfied. SB is sequential; i.e., SB selects the next input combination after analyzing the preceding I/O data. SB is customized; i.e., SB accounts for the specific simulation model. (Frazier et al. (2012) considers SB from a Bayesian dynamic programming perspective.)

4.1 Deterministic simulation and first-order polynomials

To explain the basic idea of SB, we assume a deterministic simulation and a first-order polynomial metamodel. Moreover, we replace the standardized x_j by the original simulation input z_j ($j = 1, \dots, k$), and β_j by (say) γ_j . Furthermore, we assume that the *sign* of γ_j is known; some reflection shows that we can then define the low and high bounds L_j and H_j of z_j such that $\gamma_j \geq 0$. Consequently we can rank the inputs such that the most important input has the largest γ_j . We call input j *important* if $\gamma_j > c_\gamma$ where $c_\gamma \geq 0$ is a threshold specified by the users.

In *step 1* of SB we aggregate the k inputs into *a single group*, and check whether that group is important. We therefore obtain the simulation outputs $w(\mathbf{z} = \mathbf{L})$ with $L = (L_1, \dots, L_k)'$; we also obtain $w(\mathbf{z} = \mathbf{H})$ with $\mathbf{H} = (H_1, \dots, H_k)'$. If $\gamma_j = 0$ held for all j , then our assumptions would imply $w(\mathbf{z} = \mathbf{L}) = w(\mathbf{z} = \mathbf{H})$. In practice, however, $\gamma_j > 0$ for one or more j , so $w(\mathbf{z} = \mathbf{L}) < w(\mathbf{z} = \mathbf{H})$. (If $0 \leq \gamma_j < c_\gamma$ for all j , then $w(\mathbf{z} = \mathbf{H}) - w(\mathbf{z} = \mathbf{L}) >$

c_γ may happen; this "false importance" will be discovered in the next steps, as we shall see.)

In *step 2* we *split* the group with k inputs into two subgroups: *bifurcation*. Let k_1 and k_2 denote the size of subgroup 1 and 2 respectively (obviously, $k_1 + k_2 = k$). Let $w_{[k_1]}$ (subscript within $[]$) denote w when $z_1 = H_1, \dots, z_{k_1} = H_{k_1}$ and $z_{k_1+1} = L_1, \dots, z_k = L_k$. In this step we obtain $w_{[k_1]}$, and compare this $w_{[k_1]}$ with $w_{[0]} = w(\mathbf{z} = \mathbf{L})$ of step 1: if (but not "if and only if") $w_{[k_1]} - w_{[0]} < c_\gamma$, then none of the individual inputs within subgroup 1 is important and we *eliminate* this subgroup from further experimentation. We also compare $w_{[k_1]}$ with $w_{[k]} = w(\mathbf{z} = \mathbf{H})$ of step 1; $w_k - w_{k_1} > c_\gamma$ if at least one input is important and this input is a member of subgroup 2.

In the next steps we continue splitting important subgroups into (smaller) subgroups, and eliminate unimportant subgroups. In some steps we may find that both subgroups are important; we then experiment with two important subgroups *in parallel*. Finally we identify and estimate all *individual* inputs that are not in eliminated (unimportant) subgroups.

Obviously, $\gamma_j \geq 0$ ensures that first-order effects do not cancel each other within a group. In practice the users often do know the *signs* of γ_j ; e.g., if some inputs are service speeds, then higher speeds decrease the mean waiting time. If in practice it is hard to specify the signs of a few specific inputs, then we should treat these inputs *individually* through one of the classic designs discussed in Section 2 (e.g., a R-IV design); this is safer than applying SB and assuming a negligible probability of cancellation within a subgroup.

The *efficiency* of SB—measured by the number of simulated input combinations—improves if we can label the individual inputs such that they are placed in *increasing* order of importance; this labeling implies that the important inputs are *clustered* within the same subgroup. The efficiency further improves when we place (e.g.) all service inputs in the same subgroup. Anyhow, subgroups of *equal* size are not necessarily optimal. Academic and practical examples of SB are given in Kleijnen (2015, pp. 136–172).

4.2 Random simulation and second-order polynomials

Now we assume a second-order polynomial metamodel. Moreover, we assume that if input j has $\gamma_j = 0$, then all its second-order effects $\gamma_{j;j'}$ ($j \leq j'$) are also zero: so-called *heredity* assumption. SB then gives $\hat{\gamma}_j$ unbiased by $\gamma_{j;j'}$ —provided we simulate the *mirror inputs* of the original inputs in the sequential design; see again the *foldover* principle in Section 2.2. So let $w_{[-j]}$ denote the mirror output with $z_1 = L_1, \dots, z_j = L_j$ and $z_{j+1} = H_{j+1}, \dots, z_k = H_k$. We standardize z such that x is either -1 or 1 with first-order effect β (Section 4.1 uses z instead of x). Kleijnen (2015, p. 149) shows that an unbiased estimator of $\beta_{j'-j} = \sum_{h=j'}^j \beta_h$ (the endash in $j' - j$ denotes "through") is

$$\hat{\beta}_{j'-j} = \frac{(w_{[j]} - w_{[-j]}) - (w_{[j'-1]} - w_{[-(j'-1)]})}{4}. \quad (4)$$

First we assume a *fixed* number of replications, m . Let $w_{[j;r]}$ with $r = 1, \dots, m$

denote observation r on $w_{[j]}$. Using (4), we then obtain $\hat{\beta}_{(j'-j);r}$, which gives the classic estimators of the mean and the variance of $\hat{\beta}_{(j'-j)}$. These estimated mean and variance give t_{m-1} . SB uses this t_{m-1} in a one-sided test, because $\gamma_j > 0$ so $\beta_j > 0$.

Next we assume a *random m*. Whereas the preceding t -test assumes a favorite null-hypothesis and a fixed m , the *sequential probability ratio test* (SPRT) in Wan et al. (2010) considers two comparable hypotheses and a random m in each step. This SPRT controls the type-I error probability through the whole procedure and holds the type-II error probability at each step. The users must select thresholds $c_{\gamma U}$ and $c_{\gamma I}$ such that $\beta_j \leq c_{\gamma U}$ is *unimportant* and $\beta_j \geq c_{\gamma I}$ is *important*; when $c_{\gamma U} < \beta_j < c_{\gamma I}$, the power is "reasonable". Statistical details of this SPRT and a Monte Carlo experiment are given in Kleijnen (2015, p. 154–159). (The robustness of several SPRTs is investigated in Kleijnen and Shi (2017).)

In practice, simulation models have *multiple response types*; see again Section 3.1. We can extend SB to *multiresponse SB* (MSB). Details including extensive Monte Carlo experiments, a case study concerning a logistic system in China, and the validation of the MSB assumptions are given in Kleijnen (2015, pp. 159–172) and Shi and Kleijnen (2017).

5 Kriging metamodels and their designs

Kriging metamodels are fitted to simulation I/O data obtained for *global* experimental areas, which are larger than the *local* areas in low-order polynomial metamodels. Because we assume that many readers are not familiar with the basics of Kriging, we detail various types of Kriging. We use the same symbols as above, unless Kriging traditionally uses different symbols.

5.1 Ordinary Kriging in deterministic simulation

Ordinary Kriging (OK) is popular and successful in practical deterministic simulation. OK assumes $y(\mathbf{x}) = \mu + M(\mathbf{x})$ where μ is the constant mean $E[y(\mathbf{x})]$ and $M(\mathbf{x})$ is a zero-mean Gaussian stationary process, which has covariances that depend only on the distance between the input combinations $\mathbf{x}$ and $\mathbf{x}'$. We call $M(\mathbf{x})$ the *extrinsic noise* (to be distinguished from "intrinsic" noise in stochastic simulation; see Section 5.3). Let $\mathbf{X}$ denote the $n \times k$ matrix with the n "old" combinations $\mathbf{x}_i$ ($i = 1, \dots, n$), where the original simulation inputs $\mathbf{z}_i$ are standardized to obtain $\mathbf{x}_i$ (unlike DOE, Kriging does not use the symbol $\mathbf{D}$ for the design matrix). The *best linear unbiased predictor* (BLUP) for the *new* combination $\mathbf{x}_0$ is $\hat{y}(\mathbf{x}_0) = \sum_{i=1}^n \lambda_i w_i = \boldsymbol{\lambda}' \mathbf{w}$. This unbiased $\hat{y}(\mathbf{x}_0)$ implies that if $\mathbf{x}_0 = \mathbf{x}_i$, then $\hat{y}(\mathbf{x}_0)$ is an *exact interpolator*: $\hat{y}(\mathbf{x}_i) = w(\mathbf{x}_i)$. The "best" $\hat{y}(\mathbf{x}_0)$ minimizes the *mean squared error* (MSE), which equals $\text{Var}[\hat{y}(\mathbf{x}_0)]$; see (7) below. Altogether, the *optimal* $\boldsymbol{\lambda}$ is

$$\boldsymbol{\lambda}'_o = [\sigma_M(\mathbf{x}_0) + \mathbf{1}_n' \frac{1 - \mathbf{1}_n' \boldsymbol{\Sigma}_M^{-1} \sigma(\mathbf{x}_0)}{\mathbf{1}_n' \boldsymbol{\Sigma}_M^{-1} \mathbf{1}_n}]' \boldsymbol{\Sigma}_M^{-1} \quad (5)$$

where $\mathbf{1}_n$ denotes the n -dimensional vector with ones, $\mathbf{\Sigma}_M = (\sigma_{i;i'}) = (\text{Cov}(y_i, y_{i'}))$ ($i, i' = 1, \dots, n$) denotes the $n \times n$ matrix with the covariances between the metamodel's old outputs y_i , and $\sigma_M(\mathbf{x}_0) = (\sigma_{0;i}) = (\text{Cov}(y_0, y_i))$ denotes the n -dimensional vector with the covariances between the metamodel's new output y_0 and y_i . The components of λ_o decrease with the *distance* between $\mathbf{x}_0$ and $\mathbf{x}_i$ (so λ is not a constant vector, whereas β in linear regression is). Substitution of λ_o into $\hat{y}(\mathbf{x}_0) = \mathbf{\lambda}'\mathbf{w}$ gives

$$\hat{y}(\mathbf{x}_0) = \mu + \sigma_M(\mathbf{x}_0)' \mathbf{\Sigma}_M^{-1} (\mathbf{w} - \mu \mathbf{1}_n). \quad (6)$$

Obviously, $\hat{y}(\mathbf{x}_0)$ varies with $\sigma_M(\mathbf{x}_0)$, whereas μ , $\mathbf{\Sigma}_M$, and $\mathbf{w}$ remain fixed.

Note: The *gradient* $\nabla(\hat{y})$ follows from (6); see Lophaven et al. (2002, Eq. 2.18). Sometimes we can also compute $\hat{\nabla}(w)$ and estimate a better OK model; see Kleijnen (2015, pp. 183–184) and Ulaganathan et al. (2014).

Instead of the symbol $\text{Var}(y_i) = \sigma_{i;i} = \sigma_i^2 = \sigma^2$ we use the classic Kriging symbol τ^2 :

$$\text{Var}[\hat{y}(\mathbf{x}_0)] = \tau^2 - \sigma_M(\mathbf{x}_0)' \mathbf{\Sigma}_M^{-1} \sigma_M(\mathbf{x}_0) + \frac{[1 - \mathbf{1}_n' \mathbf{\Sigma}_M^{-1} \sigma_M(\mathbf{x}_0)]^2}{\mathbf{1}_n' \mathbf{\Sigma}_M^{-1} \mathbf{1}_n}. \quad (7)$$

This equation implies $\text{Var}[\hat{y}(\mathbf{x}_0)] = 0$ if $\mathbf{x}_0 = \mathbf{x}_i$. Experimental results suggest that $\text{Var}[\hat{y}(\mathbf{x}_0)]$ has local maxima at $\mathbf{x}_0$ approximately halfway between old input combinations (also see Section 6.2). Kriging gives bad extrapolations compared with interpolations (linear regression also gives minimal $\text{Var}[\hat{y}(\mathbf{x}_0)]$ when $\mathbf{x}_0 = \mathbf{0}$).

Sometimes it is convenient to switch from covariances to correlations; the correlation matrix $\mathbf{R} = (\rho_{i;i'})$ equals $\tau^{-2} \mathbf{\Sigma}_M$ and $\rho(\mathbf{x}_0)$ equals $\tau^{-2} \sigma_M(\mathbf{x}_0)$. There are several types of correlation functions; see Kleijnen (2015, pp.185–186). Most popular is the *Gaussian correlation function*:

$$\rho(\mathbf{h}) = \prod_{j=1}^k \exp(-\theta_j h_j^2) = \exp\left(-\sum_{j=1}^k \theta_j h_j^2\right) \quad (8)$$

with distance vector $\mathbf{h} = (h_j)$ where $h_j = |x_{g;j} - x_{g';j}|$ and $g, g' = 0, 1, \dots, n$.

Obviously, we need to estimate the *Kriging (hyper)parameters* $\psi = (\mu, \tau^2, \theta')'$ with $\theta = (\theta_j)$. The most popular criterion is *maximum likelihood* (ML) (but OLS and cross-validation are also used). The computation of the *ML estimator* (MLE) $\hat{\psi}$ is challenging, so different $\hat{\psi}$ may result from different software packages or from initializing the same package with different starting values; see Erickson et al. (2017).

Plugging $\hat{\psi}$ into (6) gives $\hat{y}(\mathbf{x}_0, \hat{\psi})$. Obviously, $\hat{y}(\mathbf{x}_0, \hat{\psi})$ is a *nonlinear* predictor. In practice, we *plug* $\hat{\psi}$ into (7) to obtain $s^2[\hat{y}(\mathbf{x}_0, \hat{\psi})]$. To obtain a symmetric $(1 - \alpha)$ CI for $w(\mathbf{x}_0)$, we use $z_{\alpha/2}$ (standard symbol for the $\alpha/2$ quantile of $N(0, 1)$) and get $\hat{y}(\mathbf{x}_0, \hat{\psi}) \pm z_{\alpha/2} s[\hat{y}(\mathbf{x}_0, \hat{\psi})]$. There is much *software* for Kriging; see the many publications and websites in Kleijnen (2015, p. 190).

5.2 Designs for deterministic simulation

There is an abundant literature on various design types for Kriging in deterministic simulation; e.g., orthogonal array, uniform, maximum entropy, minimax, maximin, integrated mean squared prediction error, and "optimal" designs; see Kleijnen (2015, p. 198). However, the most popular *space filling* design uses *Latin hypercube sampling* (LHS). LHS assumes that the metamodel is more complicated than a low-order polynomial, but it does not assume a specific type of metamodel. LHS divides the range of each input into n mutually exclusive and exhaustive intervals of equal probability, and samples each input without replacement. Whereas DOE makes n increase with k (e.g., $n = 2^{k-p}$), LHS does not impose such a relationship. Nevertheless, if n is "small" and k is "large", then LHS covers the input space so sparsely that the fitted Kriging model may be inadequate. A well-known rule-of-thumb for LHS in SA through Kriging is $n = 10k$ (for SimOpt we select n sequentially; see Section 6.2). LHS may be further refined, leading to maximin LHS, nearly-orthogonal LHS, etc.; see Kleijnen (2015, p. 202).

After an initial LHS with its *fixed sample* or *one-shot* design of size (say) n_0 , we may use a *sequential* design that is *customized* for the given simulation model; i.e., we learn about f_{sim} as we collect I/O data (also see Section 4 on SB). In Kriging, we may re-estimate ψ as new I/O data become available. Next we fit a Kriging model to these data. Then we may consider—but not yet simulate— $\mathbf{X}_{\text{cand}}$ which denotes a larger matrix with *candidate* combinations selected through LHS, and find the candidate with the highest $s^2\{\hat{y}(\mathbf{x}, \psi)\}$ with $\mathbf{x} \in \mathbf{X}_{\text{cand}}$. Next we use the selected candidate point as the input to be simulated, which gives additional I/O data. Then we re-fit the Kriging model to the augmented I/O data. Finally we stop if either the Kriging metamodel satisfies a given goal or the computer budget is exhausted. Experiments show that such a design selects relatively few combinations in subareas with an approximately linear f_{sim} . Details are given in Kleijnen (2015, pp. 203–206).

5.3 Kriging in random simulation

Ankenman et al. (2010) develops *stochastic Kriging* (SK), adding the *intrinsic noise* term $\varepsilon_r(\mathbf{x}_i)$ for replication r ($r = 1, \dots, m_i$) at combination $\mathbf{x}_i$ ($i = 1, \dots, n$). After averaging over the m_i replications, SK uses the formulas for OK but replaces $\mathbf{w}$ by $\bar{\mathbf{w}}$ and $M(\mathbf{x}_i)$ by $M(\mathbf{x}_i) + \bar{\varepsilon}(\mathbf{x}_i)$ where $\bar{\varepsilon}(\mathbf{x}_i) \in N(0, \text{Var}[\varepsilon_r(\mathbf{x}_i)]/m_i)$ and $\bar{\varepsilon}(\mathbf{x}_i)$ is independent of $M(\mathbf{x})$. If we do not use CRN, then $\Sigma_{\bar{\varepsilon}}$ is diagonal (if we used CRN and selected $m_i = m$, then $\Sigma_{\bar{\varepsilon}}$ would equal Σ_{ε}/m ; however, we assume no CRN in this subsection). To estimate $\text{Var}[\varepsilon(\mathbf{x}_i)]$, SK may use either $s^2(w_i)$ or another Kriging model for $\text{Var}[\varepsilon(\mathbf{x}_i)]$ —besides the Kriging model for the mean $E[y_r(\mathbf{x}_i)]$; see Kleijnen (2015, p.208). We use $\psi_{+\varepsilon}$ to denote ψ augmented with $\text{Var}[\varepsilon_r(\mathbf{x}_i)]$ (extra parameters of random simulation). Note that SK for a quantile instead of an average is discussed in Chen and Kim (2013).

Binois et al. (2016) gives an alternative—called *hetGP*—for SK. This alternative assumes $m_i \geq 1$, whereas SK assumes $m_i \gg 1$. SK maximizes a

likelihood for ψ , but not the full likelihood; i.e., SK does not give the MLEs of the intrinsic variances, but uses the moments $s^2(w_i)$. So, $\hat{\psi}_{+\varepsilon}$ in SK is biased. The alternative uses the full likelihood for $\psi_{+\varepsilon}$; i.e., whereas SK fits the Kriging models for the mean and the intrinsic variances independently, hetGP couples these models through a joint likelihood for $\psi_{+\varepsilon}$ that is optimized in one shot. This alternative requires computational time of the same order as SK does.

5.4 Monotonic Kriging

In practice we sometimes know that f_{sim} is *monotonic*; also see Section 4 on screening. The Kriging predictor $\hat{y}$, however, may be *wiggling* if the sample size is small. To make $\hat{y}$ monotonic, we may apply *distribution-free bootstrapping* with *acceptance/rejection*; see Kleijnen et al. (2012). This method rejects the Kriging model fitted in bootstrap sample b ($b = 1, \dots, B$) if this metamodel gives wiggling predictions. We think that the resulting SA is better understood and accepted by the users. Monotonic Kriging may give a smaller MSE and a CI with higher coverage and acceptable length. For details we refer to Kleijnen (2015, pp. 212–216).

We may apply this method to preserve other I/O characteristics; e.g., waiting times and variances have only *positive* values. Furthermore, we may apply this method to other types of metamodels; e.g., regression models.

5.5 Global sensitivity analysis

So far we focused on $\hat{y}$, but now we measure how sensitive $\hat{y}$ and w are to the individual inputs x_j ($j = 1, \dots, k$) and their interactions. We assume that $x = (x_j)$ has a prespecified distribution, as in LHS; see Section 5.2. We apply *functional analysis of variance* (FANOVA), using variance-based indexes (originally proposed by Sobol). FANOVA decomposes σ_w^2 into fractions that refer to sets of inputs; e.g., FANOVA may estimate that 70% of σ_w^2 is caused by x_1 , 20% by x_2 , and 10% by the interaction between x_1 and x_2 . In general, σ_w^2 equals the following sum of 2^{k-1} components:

$$\sigma_w^2 = \sum_{j=1}^k \sigma_j^2 + \sum_{j < j'}^k \sigma_{j;j'}^2 + \dots + \sigma_{1;\dots;k}^2 \quad (9)$$

with the main-effect variance $\sigma_j^2 = \text{Var}[E(w|x_j)]$, the two-factor interaction variance $\sigma_{j;j'}^2 = \text{Var}[E(w|x_j, x_{j'})] - \text{Var}[E(w|x_j)] - \text{Var}[E(w|x_{j'})]$, etc. This σ_j^2 gives the *first-order sensitivity index* or the *main effect index* $\zeta_j = \sigma_j^2/\sigma_w^2$, which quantifies the effect of varying x_j alone—averaged over the variations in all the other $k-1$ inputs—where the denominator σ_w^2 standardizes ζ_j to provide a fractional contribution. Altogether the sum of the 2^{k-1} indexes is 1. In practice, we assume that only the ζ_j —and possibly the $\zeta_{j;j'}$ —are important, and that they sum up to a fraction "close enough" to 1. To *estimate* these measures, we may use LHS and replace the simulation model by a Kriging metamodel; see Kleijnen (2015, p. 218).

5.6 Risk analysis or uncertainty analysis

In RA (or UA) we typically estimate the *failure probability* $P(w > c_f)$ with a given threshold value c_f ; see the many applications in nuclear engineering, finance, water management, etc. This $P(w > c_f)$ may be very small—so $w > c_f$ is called a *rare event*—but this event may have disastrous consequences. Spreadsheets are popular software for such RA. The uncertainty about the exact values of the simulation inputs $\mathbf{z}$ is called *subjective* or *epistemic*, whereas the "intrinsic" uncertainty in random simulation is called *objective* or *aleatory*.

SA and RA address different questions: "Which are the most important inputs in the simulation model?" and "What is the probability of a given event happening?" respectively. So, SA may identify those inputs for which the distribution in RA needs further refinement.

Methodologically, we use a Monte Carlo method to sample input combination $\mathbf{z}$ from its given distribution. Next we use this $\mathbf{z}$ as input into the given simulation model. We run the simulation model to transform $\mathbf{z}$ into w : so-called *propagation of uncertainty*. We repeat these steps "many" times to obtain the EDF of w . Finally, we use this EDF to estimate $P(w > c_f)$. This method is also known as *nested simulation*; see Kleijnen (2015, p. 284).

In *expensive* simulation, we replace the simulation model by its *metamodel*. The true $P(w > c_f)$ may be better estimated through inexpensive sampling of many values from the metamodel, which is estimated from relatively few I/O values obtained from the expensive simulation model.

Instead of $P(w > c_f)$ we may estimate the *excursion set*, which is the set of input combinations that give an output that exceeds a given threshold. Obviously, the volume of the excursion set is closely related to $P(w > c_f)$. Details on RA though Kriging and regression metamodels are given in Kleijnen (2015, p. 221) and Nelson and Song (2017).

6 Simulation optimization

The importance of optimization is demonstrated by the many WSC publications on this topic. The simplest optimization problem has no constraints for the inputs or the outputs, has no uncertain inputs, and concerns the expected value of a single output, $E(w)$. This $E(w)$ may represent the probability of a binary variable. However, $E(w)$ excludes quantiles and the mode of the output distribution. Furthermore, the simplest optimization problem assumes continuous inputs, excluding MRP and MCP (also see Section 1).

There are many optimization methods; also see Rzyhlov (2017). We, however, focus RSM (using linear-regression) and Kriging metamodels. Jalali and Van Nieuwenhuyse (2015) claims that metamodel-based optimization is "relatively common" and that RSM is the most popular metamodel-based method, while Kriging is popular in theoretical publications. We add that in *expensive* simulation it is impractical to apply optimization methods such as the popular *OptQuest*. A single simulation run may be computationally inexpensive, but

there may be extremely many input combinations. Furthermore, most simulation models have many inputs, which leads to the *curse of dimensionality*, so we should apply screening before optimization. Moreover, a single run is expensive if we want an accurate estimate of the steady-state performance of a queueing system with a high traffic rate. Finally, if we wish to estimate a small $E(w) = p$ (e.g. $p = 10^{-7}$), then we need extremely long simulation runs (unless we successfully apply importance sampling).

6.1 Response surface methodology

RSM uses a sequence of local experiments, and has gained a good track record; see Kleijnen (2015, p. 244), Law (2015, pp. 656–679), and Myers et al. (2009). We assume that before RSM is applied, we have identified the important inputs and their experimental area (RSM and screening may be combined; see Kleijnen (2015, p. 245)).

Methodologically, RSM tries to minimize $E(w|\mathbf{z})$. We start RSM in a given point; e.g., the combination currently used in practice. In the *neighborhood* of this point we fit a first-order polynomial, assuming white noise; however, RSM allows $\text{Var}(w)$ to change in a next step. Unfortunately, there are no general guidelines for determining the appropriate *size* of the local area in each step. To fit a first-order polynomial, we use a *R-III design* (see Section 2.1). To quantify the *adequacy* of this estimated polynomial, we may compute R^2 or cross-validation statistics; see Section 3.5. In the next steps, we use the *gradient* implied by this first-order polynomial: $\nabla(\hat{y}) = \hat{\gamma}_{-0}$ where -0 means that the intercept $\hat{\gamma}_0$ is removed from the $(k+1)$ -dimensional vector with the estimated regression parameters $\hat{\gamma}$. This $\nabla(\hat{y})$ implies the *steepest descent* direction. We take a step in that direction, trying intuitively selected values for the *step size*. After a number of such steps, w will increase (instead of decrease) because the *local* first-order polynomial becomes inadequate. When such deterioration occurs, we simulate the $n > k$ combinations of the R-III design—but now centered around the best combination found so far. We re-estimate the polynomial and in the resulting new steepest-descent direction we again take several steps. Obviously, a *plane* (implied by a first-order polynomial) cannot adequately represent a *hill top* when searching to maximize w or—equivalently—minimize w . So, in the neighborhood of the latest estimated optimum we now fit a *second-order* polynomial, using a CCD. Next we use the derivatives of this polynomial to estimate the optimum; we may apply *canonical analysis* to examine the shape of the estimated optimal subregion: does this subregion give a unique minimum, a saddle point, or a ridge with stationary points? If time permits, then we may try to escape from a possible local optimum and *restart* the search from a different initial local area.

While applying RSM, we should not eliminate inputs with *nonsignificant* effects in a local first-order polynomial: these inputs may become significant in a next local area. The selection of the number of replications in RSM for random simulation is a moot issue; see Section 3.3 and the SPRT for SB in Section 4.2.

After discussing classic RSM, we summarize three extensions. (i) A *scale-independent* steepest-descent direction that accounts for $\Sigma_{\hat{\gamma}}$ is discussed in Kleijnen (2015, pp. 252–253). (ii) In practice, simulation models have *multiple* responses types (see again Sections 3.1 and 4.2). For such situations the RSM literature offers several approaches, but we focus on *generalized RSM* (GRSM), which solves the following *constrained nonlinear random optimization problem*:

$$\begin{aligned} \min_{\mathbf{z}} E(w^{(1)}|\mathbf{z}) \\ E(w^{(l')})|\mathbf{z} \geq c_{l'} \quad (l' = 2, \dots, r) \\ L_j \leq z_j \leq H_j \quad \text{with } j = 1, \dots, k. \end{aligned} \quad (10)$$

GRSM combines RSM and *interior point* methods from MP, avoiding creeping along the boundary of the feasible area that is determined by the constraints on the random outputs and the deterministic inputs. So, GRSM moves faster to the optimum, and is scale independent. For details we refer to Kleijnen (2015, pp. 253–258). (iii) Because the GRSM heuristic may miss the true optimum, we can test the first-order necessary optimality or *Karush - Kuhn - Tucker* (KKT) conditions. To test these conditions, we may use *parametric bootstrapping* that samples w^* from the assumed distribution; namely, a multivariate normal) with parameters estimated from the original w . Details are given in Kleijnen (2015, pp. 259–266).

6.2 Kriging for optimization

Kriging is used by *efficient global optimization* (EGO), which is a popular *sequential* method that balances *local* and *global* search; i.e., EGO balances *exploitation* and *exploration*. We present only the *basic* EGO-variant for *deterministic* simulation; also see the classic EGO reference, Jones et al. (1998). There are many more variants for deterministic and random simulations, constrained optimization, multi-objective optimization including Pareto frontiers, the "excursion set" or "admissible set", robust optimization, estimation of a quantile, and Bayesian approaches; see Kleijnen (2015, p. 267–269).

In this basic variant we consider the best simulated "old" output: $f_{\min} = \min_i w(\mathbf{x}_i)$ ($i = 1, \dots, n$). To select a new combination $\mathbf{x}_0$, we consider $\hat{y}(\mathbf{x}_0)$ and $s^2[\hat{y}(\mathbf{x})]$ (we suppress $\hat{\psi}$); e.g., if two new points $\mathbf{x}_0$ and $\mathbf{x}'_0$ have $\hat{y}(\mathbf{x}_0) = \hat{y}(\mathbf{x}'_0)$ and $s^2[\hat{y}(\mathbf{x}_0)] > s^2[\hat{y}(\mathbf{x}'_0)]$, then we explore $\mathbf{x}_0$ because $\mathbf{x}_0$ has a higher probability of improvement (lower w). We know that $s^2[\hat{y}(\mathbf{x}_0)]$ increases as $\mathbf{x}_0$ lies farther away from $\mathbf{x}_i$; see (7). Actually, we estimate the maximum of the *expected improvement* (EI), which is reached if either $\hat{y}(\mathbf{x}_0)$ is much smaller than $f_{\min}$ or $s^2[\hat{y}(\mathbf{x}_0)]$ is relatively large so $\hat{y}(\mathbf{x}_0)$ is relatively uncertain.

More precisely, we start with a pilot sample—typically selected through LHS—which results in $(\mathbf{X}, \mathbf{w})$. Next we find $f_{\min} = \min_{1 \leq i \leq n} w(\mathbf{x}_i)$. We also fit a Kriging metamodel $\hat{y}(\mathbf{x})$. Together, this gives $\text{EI}(\mathbf{x}) = E[\max(f_{\min} - \hat{y}(\mathbf{x}), 0)]$. Jones et al. (1998) derives the EI estimator

$$\widehat{\text{EI}}(\mathbf{x}) = (f_{\min} - \hat{y}(\mathbf{x})) \Phi \left(\frac{f_{\min} - \hat{y}(\mathbf{x})}{s[\hat{y}(\mathbf{x}_0)]} \right) + s[\hat{y}(\mathbf{x}_0)] \phi \left(\frac{f_{\min} - \hat{y}(\mathbf{x})}{s[\hat{y}(\mathbf{x}_0)]} \right) \quad (11)$$

where Φ and ϕ denote the cumulative distribution function (CDF) and the PDF of $N(0, 1)$. Using (11), we find the estimate of $\mathbf{x}$ that maximizes $\widehat{\text{EI}}(\mathbf{x})$ (say) $\widehat{\mathbf{x}}_{opt}$. Actually, to find $\widehat{\mathbf{x}}_{opt}$, we may use either a global optimizer or a set of candidate points selected through LHS. Next we use this $\widehat{\mathbf{x}}_{opt}$ to obtain $w(\widehat{\mathbf{x}}_{opt})$. Then we fit a new Kriging model to the augmented I/O data. We update n , and return to (11)—until we satisfy a stopping criterion; e.g., $\widehat{\text{EI}}(\widehat{\mathbf{x}}_{opt})$ is “close” to 0. Note that EGO is combined with bootstrapping in Kleijnen and Mehdad (2013).

Now we consider the constrained optimization problem in (10), augmented with s constraints f_g for $\mathbf{z}$ (e.g., budget constraints) and the constraint that z must belong to the set of nonnegative integers $\mathbf{N}$. To solve this problem, we might apply an EGO variant, as mentioned above. Alternatively, we may apply *Kriging and integer mathematical programming* (KIMP), which combines (i) *sequentialized* designs to specify the next combination, like EGO does; (ii) *Kriging* to analyze the resulting I/O data, and obtain explicit functions for $E(w^{(l)}|z)$ ($l = 1, \dots, r$), like EGO does; (iii) *integer nonlinear programming* (INLP) to estimate the optimal solution from these explicit Kriging models, unlike EGO. Experiments with KIMP and OptQuest suggest that KIMP requires fewer simulated combinations and gives better estimated optima.

6.3 Robust optimization

The estimated optimum (see the preceding two subsections) may turn out to be inferior because it ignores uncertainties in the noncontrollable environmental variables; i.e., these uncertainties create a *risk* (also see RA in Section 5.6). *Taguchians* emphasize that in practice some inputs of a manufactured product (e.g., a car) are under complete control of the engineers (the car’s design), whereas other inputs are not (the driver). Taguchians therefore distinguish between (i) *controllable* (decision) variables, and (ii) *noncontrollable* (environmental, noise) factors. Let the first k_C of the k simulated inputs be controllable, and the next k_{NC} inputs be noncontrollable. Let $\mathbf{z}_C$ and $\mathbf{z}_{NC}$ denote the vector with the k_C controllable and the k_{NC} noncontrollable inputs.

Methodologically, Taguchians assume a single output (say) w , focusing on its mean $E(w)$ and its variance, which is caused by $\mathbf{z}_{NC}$ so $\sigma^2(w|\mathbf{z}_C) > 0$. For brevity’s sake we denote $\sigma^2(w|\mathbf{z}_C)$ by σ_w^2 . These two outputs are combined into a *scalar loss function* such as the *signal-to-noise* or *mean-to-variance* ratio $E(w)/\sigma_w^2$; see Myers et al. (2009, pp. 486–488). We, however, prefer to use $E(w)$ and $\sigma(w|\mathbf{z}_C)$ (obviously, $\sigma(w|\mathbf{z}_C)$ has the same scale as $E(w)$ has) *separately* so we can use *constrained optimization*; i.e., given an upper threshold c_σ for $\sigma(w|\mathbf{z}_C)$, we try to solve $\min_{\mathbf{z}_C} E(w|\mathbf{z}_C)$ such that $\sigma(w|\mathbf{z}_C) \leq c_\sigma$. Constrained optimization is also discussed in Myers et al. (2009, p. 492).

The Taguchian worldview is successful in production engineering, but statisticians criticize the statistical techniques. Moreover—compared with real-life experiments—simulation experiments have more inputs, more input values, and more input combinations; see again Section 1. Myers et al. (2009, pp. 502–506) combines the Taguchian worldview with the statisticians’ RSM. Whereas Myers

et al. (2009) assumes that $\mathbf{z}_{\text{NC}}$ has $\Sigma_{\text{NC}} = \sigma_w^2 I$, we assume a general Σ_{NC} . Whereas Myers et al. (2009) superimposes contour plots for $E(w|\mathbf{z}_C)$ and σ_w^2 to estimate the optimal $\mathbf{z}_C$, we use MP. This MP, however, requires specification of the threshold c_σ . In practice, managers may find it hard to select a specific value for c_σ , so we may try different c_σ values and estimate the corresponding *Pareto-optimal* efficiency frontier. To estimate the variability of this frontier that results from the estimators of $E(w|\mathbf{z}_C)$ and $\sigma(w|\mathbf{z}_C)$, we may use bootstrapping. Instead of RSM we may apply Kriging. Details are given in Kleijnen (2015, pp. 273–284).

Finally, we summarize *RO in MP*; see again Bertsimas and Mišić (2017). If MP ignores the uncertainty in the coefficients of the MP model, then the resulting so-called *nominal solution* may easily violate the constraints in the given model. RO may give a slightly worse value for the goal variable, but RO increases the probability of satisfying the constraints; i.e., a robust solution is "immune" to variations of the variables within the so-called *uncertainty set* U . Yanikoğlu et al. (2016) derives a specific U for $\mathbf{p}$ where $\mathbf{p}$ denotes the unknown density function of $\mathbf{z}_{\text{NC}}$ that is compatible with the given historical data on $\mathbf{z}_{\text{NC}}$. RO in MP develops a computationally tractable *robust counterpart* of the original problem. Compared with the output of the nominal solution, RO may give better worst-case and average outputs.

References

- Barton, R. R. (1997). Design of experiments for fitting subsystem metamod-els. *Proceedings of the 1997 Winter Simulation Conference*, pp. 303–310
- Bertsimas, D. and V.V. Mišić (2017) Robust Product Line Design. *Operations Research*, 65, no. 1, pp. 19–37
- Biles, B. E., W. C. M. van Beers, J. P. C. Kleijnen, and I. van Nieuwen-huyse (2007). Kriging metamod-els in constrained simulation optimization: An exploratory study. *Proceedings of the 2007 Winter Simulation Conference*, pp. 355–362
- Binois, M., R.B. Gramacy and M. Ludkovskiz (2016), Practical heteroskedas-tic Gaussian process modeling for large simulation experiments. *ArXiv*, 17 Nov 2016
- Chen X and K-K. Kim (2013), Building metamod-els for quantile-based mea-sures using sectioning. *Proceedings of the 2013 Winter Simulation Conference*, pp 521–532
- Dellino, G., Kleijnen, J. P. C., & Meloni, C. (2009). Robust simulation-optimization using metamod-els. *Proceedings 2009 Winter Simulation Confer-ence*, pp. 540–550
- Dellino, G., J. P. C. Kleijnen, and C. Meloni (2010). Parametric and distribution-free bootstrapping in robust simulation optimization. *Proceedings 2010 Winter Simulation Conference*, pp. 1283–1294
- Erickson, C.B., S.M. Sanchez, and B.E. Ankenman (2016), Comparison of Gaussian process modeling software. *WSC 2016 Proceedings*, pp. 3692–3693

- Frazier PI, Jedynek B, Chen L (2012) Sequential screening: a Bayesian dynamic programming analysis. *Proceedings 2012 Winter Simulation Conference*, Berlin, pp. 555–566
- Govindan, E. (2016), *Yosida Approximations of Stochastic Differential Equations in Infinite Dimensions and Applications*. Springer
- Gramacy, R.B. (2015), laGP: large-scale spatial modeling via local approximate Gaussian processes in R. *Journal of Statistical Software* (available as a vignette in the laGP package)
- Jalali H. and I. Van Nieuwenhuysen (2015), Simulation optimization in inventory replenishment: a classification. *IIE Transactions*, 47, no. 11, pp. 1217–1235
- Jones, D.R., M. Schonlau, and W.J. Welch (1998), Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13, pp. 455–492
- Kleijnen, J. P. C. (1995). Case study: Statistical validation of simulation models. *European Journal of Operational Research*, 87, no. 1, pp. 21–34
- Kleijnen, J.P.C. (2015), *Design and Analysis of Simulation Experiments; Second Edition*; Springer
- Kleijnen, J. P. C. (2017), Regression and Kriging metamodels with their experimental designs in simulation: a review. *European Journal of Operational Research*, 256, pp. 1–16
- Kleijnen, J. P. C., and E. Mehdad (2013). Conditional simulation for efficient global optimization. *Proceedings 2013 Winter Simulation Conference*, pp. 969–979
- Kleijnen, J. P. C., E. Mehdad, and W.C.M. van Beers (2012). Convex and monotonic bootstrapped kriging. *Proceedings of the 2012 Winter Simulation Conference*, pp. 543–554
- Kleijnen, J. P. C. and W. Shi (2017), Sequential probability ratio tests: conservative and robust. CentER Discussion Paper; Vol. 2017-001, Tilburg: CentER, Center for Economic Research
- Law, A.M. (2015), *Simulation modeling and analysis, 5th edition*. McGraw-Hill, Boston
- Liu, Z., D. Rexachs, F. Epelde, and E. Luque (2017), A simulation and optimization based method for calibrating agent-based emergency department models under data scarcity. *Computers & Industrial Engineering*, 103, pp. 300–309
- Lophaven, S.N., H.B. Nielsen, and J. Sondergaard (2002), DACE: a Matlab Kriging toolbox, version 2.0. IMM Technical University of Denmark, Kongens Lyngby
- Mitchell, T.J. and Morris, M.D. (1992), The spatial correlation function approach to response surface estimation. *Proceedings of the 2012 Winter Simulation Conference*, pp. 565–571
- Montgomery, D. C. (2009). *Design and Analysis of Experiments; 7th Edition*, Wiley, Hoboken, NJ
- Myers, R.H., D.C. Montgomery, and C.M. Anderson-Cook (2009), *Response Surface Methodology: Process and Product Optimization Using Designed Experiments; Third Edition*. Wiley, New York

- Nelson, B.L. and E. Song (2017), Input model risk: where we stand and where we are going. *Advances in Modeling and Simulation: Seminal Research from 50 Years of Winter Simulation Conferences*, edited by A. Tolk, J. Fowler, G. Shao, and E. Yücesan
- Rzyhiov, I. (2017), Bayesian belief models in simulation-based optimization. *Advances in Modeling and Simulation: Seminal Research from 50 Years of Winter Simulation Conferences*, edited by A. Tolk, J. Fowler, G. Shao, and E. Yücesan
- Sargent, R.G., D.M. Goldsman, and T. Yaacoub (2016), A tutorial on the operational validation of simulation models. *Proceedings of the 2016 Winter Simulation Conference*, pp. 163–177
- Simpson T.W., A.J. Booker, D. Ghosh, A.A. Giunta, P.N. Koch, R-J. Yang (2004), Approximation methods in multidisciplinary analysis and optimization: a panel discussion. *Structural and Multidisciplinary Optimization*, 27, no, 5, pp. 302–313
- Shi, W. and J.P.C. Kleijnen (2017), Testing the assumptions of sequential bifurcation in factor screening. Discussion Paper; vol. 2017-06, Center for Economic Research, Tilburg
- Uhrmacher, A., S. Brailsford, J. Liu, M. Rabe, and A. Tolk (2016), Panel - Reproducible research in discrete event simulation - A must or rather a maybe? *Proceedings of the 2016 Winter Simulation Conference*, pp. 1301–1315
- Ulaganathan, S., I. Couckuyt, T. Dhaene, and E. Laermans (2014), On the use of gradients in Kriging surrogate models. *Proceedings of the 2014 Winter Simulation Conference*, pp. 2692–2701
- Van Beers, W. C. M. and J.P.C. Kleijnen (2004). Kriging interpolation in simulation: A survey. *Proceedings of the 2004 Winter Simulation Conference*, pp. 113–121
- Van Ham, G., J. Rotmans, and J.P.C. Kleijnen (1992). Techniques for sensitivity analysis of simulation models: A case study of the CO2 greenhouse effect. *Simulation*, 58, no. 6, pp. 410–417
- Wagner, H. (1975), Principles of Operations Research; with Applications to Managerial Decisions; Second Edition. Prentice-Hall, Inc., Englewood Cliffs, New Jersey
- Wan, H., B.E. Ankenman, and B.L. Nelson (2010), Improving the efficiency and efficacy of controlled sequential bifurcation for simulation factor screening. *INFORMS Journal on Computing*, 22. no. 3, pp. 482–492
- Yanikoğlu, I., den Hertog, D. & Kleijnen, J. P. C. (2016), Robust dual-response optimization. *IIE Transactions: Industrial Engineering Research and Development*, 48, no. 3, pp. 298–312
- Zeigler B.P., H. Praehofer, T.G. Kim (2000), *Theory of Modeling and Simulation, 2nd Edition*. Academic, San Diego