Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

DESIGN OF EXPERIMENTS: OVERVIEW

Jack P.C. Kleijnen

Tilburg University
Faculty of Economics and Business Administration
Tilburg, THE NETHERLANDS

ABSTRACT

Design Of Experiments (DOE) is needed for experiments with real-life systems, and with either deterministic or random simulation models. This contribution discusses the different types of DOE for these three domains, but focusses on random simulation. DOE may have two goals: sensitivity analysis and optimization. This contribution starts with classic DOE including 2^{k-p} and Central Composite Designs (CCDs). Next, it discusses factor screening through Sequential Bifurcation. Then it discusses Kriging including Latin Hypercube Sampling and sequential designs. It ends with optimization through Generalized Response Surface Methodology and Kriging combined with Mathematical Programming, including Taguchian robust optimization.

1 INTRODUCTION

DOE is needed for experiments with

- real-life (physical) systems;
- deterministic simulation models;
- random (stochastic) simulation models.

For *real-life* systems the scientific DOE—based on mathematical statistics—started with agricultural experiments in the 1920s (Sir Ronald Fisher), followed by chemical experiments in the 1950s (George Box), and is now also applied in social systems such as educational and service systems. This domain is covered extensively by Montgomery (2009) and Myers and Montgomery (1995).

In *deterministic simulation*, DOE gained popularity with the increased use of 'computer codes' for the design (in an engineering, not a statistical sense) of airplanes, automobiles, TV sets, chemical processes, computer chips, etc.—in Computer Aided Engineering (CAE) and Computer Aided Design (CAD)—at companies such as Boeing, General Motors, and Philips; see Koehler and Owen (1996), Santner, Williams, and Notz (2003), and also Kleijnen (008a). This domain often does not use the term DOE but *DACE*, Design and Analysis of Computer Experiments.

Random simulation includes 'Discrete-Event Dynamic Systems (DEDS)' such as queuing and inventory models, but also stochastic difference and differential equation models. This type of simulation is the focus of the yearly Winter Simulation Conference (WSC). DOE for random simulation is the focus of Kleijnen (008a) and of this overview.

DOE may vary with the type of experiment. In real-life experiments it is not practical to investigate many factors; ten factors seems a maximum. Moreover, in these experiments it is hard to experiment with many values (or 'levels') per factor; five values per factor seems the limit. In experiments with simulation models (either deterministic or random), however, these restrictions do not apply. Indeed, computer codes may have hundreds of inputs and parameters—each with many values. Consequently, a multitude of *scenarios*—combinations of factor values—may be simulated. Moreover, simulation is well-suited to sequential designs instead of 'one shot' designs (ignoring simulation on parallel computers). So a change of mindset of simulation experimenters is necessary; see Kleijnen et al. (2005).

Random (unlike deterministic) simulation uses *Pseudo-Random Numbers* (PRNs) inside its model; e.g., a queueing simulation uses random service times (say, exponentially distributed). *Common pseudo-Random Numbers (CRN)* are often used when simulating different input combinations; e.g., the popular simulation software called 'Arena' uses CRN as its default when simulating different scenarios. CRN violate the classic DOE assumption of *white noise*, because CRN make the simulation outputs (responses) positively correlated instead of independent.

DOE for real-life experiments pays much attention to *blocked designs*, because the environment cannot be controlled, which creates undesirable effects such as learning curves. In simulation experiments, such effects do not occur, because everything is completely under control—except for the PRNs. CRN and antithetic PRN can be used as a block

978-1-4244-2708-6/08/$25.00 ©2008 IEEE

factor in simulation; see Schruben and Margolin (1978) and also Kleijnen (008a).

DOE for real-life experiments often uses *fractional factorial designs* such as 2^{k-p} designs: each of the k factors has only two values and of all the 2^k combinations only 2^{k-p} combinations are observed; e.g., a 2^{7-4} design means that of all $2^7 = 128$ combinations only a $2^{-4} = 1/16$ fraction is executed. This 2^{7-4} design is acceptable if the experimenters assume that a first-order polynomial is an adequate approximation or—as we say in simulation— a valid 'metamodel'. A *metamodel* is an approximation of the Input/Output (I/O) function implied by the underlying simulation model. Besides first-order polynomials, classic designs may also assume a first-order ('main effects') metamodel augmented with the interactions between pairs of factors, among triplets of factors, ..., and 'the' interaction among all the k factors (however, I am against assuming such high-order interactions, because they are hard to interpret). Moreover, classic DOE may assume a second-order polynomial. See Montgomery (2009), Myers and Montgomery (1995), and also Kleijnen (008a).

In deterministic simulation, another metamodel type is popular, namely *Kriging* (also called spatial correlation or Gaussian) models. Kriging is an exact interpolator; i.e., for 'old' simulation input combinations the Kriging prediction equals the observed simulation outputs—which is attractive in deterministic simulation. Because Kriging has just begun in random simulation, I will discuss this type of metamodel in more detail; see Section 4.

Each type of metamodel requires a different design type, and vice versa: chicken-and-egg problem. Therefore I proposed the term *DASE*, Design and Analysis of Simulation Experiments, in Kleijnen (008a). Which design/metamodel is acceptable is determined by the goal of the simulation study. Different goals are considered in the methodology for the validation of metamodels presented in Kleijnen and Sargent (2000). I focus on two goals:

- Sensitivity Analysis (SA);
- optimization.

SA may serve *Validation & Verification* (V & V) of simulation models, and *factor screening*—or briefly screening—which denotes the search for the really important factors among the many factors that are varied in an experiment. Optimization tries to find the optimal combination of the decision factors in the simulated system. Optimization may follow after SA. Recently, I have become interested in *robust* optimization, which assumes that the environmental factors (not the decision factors) are uncertain.

The remainder of this contribution is organized as follows. Section 2 presents classic designs and the corresponding metamodels. Section 3 reviews screening, focussing on Sequential Bifurcation (SB). Section 4 reviews Kriging and its designs. Section 5 discusses simulation optimization, focussing on Generalized Response Surface methodology (GRSM), Kriging combined with Mathematical Programming (MP), and Taguchian robust optimization. Section 6 presents conclusions. This overview is based on my recent book, Kleijnen (008a) and some of my recent papers; see the References at the end of this contribution.

2 CLASSIC DESIGNS AND METAMODELS

In this section, I do not discuss in detail classic designs and their corresponding metamodels, because these designs and metamodels are discussed in many DOE textbooks such as Montgomery (2009) and Myers and Montgomery (1995); these designs and models are presented from a simulation perspective in Kleijnen (008a). I do give a simple example, and discuss the classic assumptions of univariate output and white noise.

1. Resolution-III (R-III) designs for first-order polynomials, which include Plackett-Burman and 2^{k-p} designs;
2. Resolution-IV (R-IV) and resolution-V (R-V) designs for two-factor interactions;
3. designs for second-degree polynomials, which include CCDs.

I illustrate these various designs through the following example with $k = 6$ factors.

1. To estimate the first-order polynomial metamodel, obviously at least $k+1 = 7$ combinations are needed. The eight combinations of a 2^{7-4} design ignoring the column for factor 7 enable the Ordinary Least Squares (OLS) estimation of the first-order effects (say) β_j $(j = 1,\ldots,6)$ and the intercept β_0. OLS is the classic estimation method in linear regression analysis, assuming white noise.
2. A R-IV design would ensure that these estimated first-order effects are not biased by the two-factor interactions $\beta_{j;j'}$ $(j < j' = 2,\ldots 6)$. However, to estimate the $k(k-1)/2 = 15$ individual interactions, a R-V design is needed. A 2^{6-1} design is a R-V design, but its 32 combinations take too much computer time if the simulation model is computationally expensive. In that case, Rechtschaffner's *saturated* design is better; see Kleijnen (2008a, p.49); by definition a saturated design has a number of combinations (say) n that equals the number of metamodel parameters (say) q.
3. A CCD for the second-degree polynomial enables the estimation of the k 'purely quadratic effects' $\beta_{j;j}$. Such a CCD augments the R-V design with the 'central point' of the experimental area and $2k$

'axial points', which change each factor one-at-a-time by $-c$ and c units where $c > 0$. Obviously the CCD is rather wasteful in case of expensive simulation, because it has five values per factor (instead of the minimum, three) and it is not saturated. Alternatives for the CCD are discussed in Kleijnen (008a) and Myers and Montgomery (1995).

The assumptions of these classic designs and metamodels stipulate univariate output and white noise. Kleijnen (008a) and Kleijnen (008b) discuss the following.

1. *Multivariate* (multiple) simulation output may still be analyzed through OLS because Generalized Least Squares (GLS) reduces to OLS if the multiple outputs use the same input combinations (same design).
2. *Nonnormality* of the simulation output may be extreme (e.g., a small probability is estimated), in which case it may be tackled through either jackknifing or bootstrapping. The basics of bootstrapping are explained in Efron and Tibshirani (1993) and Kleijnen (008a).
3. *Variance heterogeneity* may be addressed through Estimated Weighted Least Squares (EWLS) using estimated variances, which results in a nonlinear estimator so either jackknifing or bootstrapping may be applied.
4. *CRN* creates correlation between the outputs of input combinations; the OLS estimate of the factor effects may be computed per replicate so the analysis is straightforward if there are at least two replicates per factor combination.
5. The *validity* of low-order polynomial metamodels may be tested through either the classic F lack-of-fit statistic or the popular cross-validation method.

3 SCREENING: SEQUENTIAL BIFURCATION (SB)

SB was originally published back in 1990; see Bettonvil (1990). SB is most efficient and effective if its assumptions are indeed satisfied. This section summarizes SB, including its assumptions. This section also references recent research. It ends with a discussion of possible topics for future research. This section is based on Kleijnen (008a) and Kleijnen (008c), which also reference other screening methods besides SB. Recently, SB has attracted the attention of several researchers in the UK and USA; see Xu, Yang, and Wan (2007). Notice that some authors call R-III designs (discussed in Section 2) screening designs; see Yu (2007).

Screening is related to 'sparse' effects, the 'parsimony' or 'Pareto' principle, 'Occam's razor', the '20-80 rule', the 'curse of dimensionality', etc. Practitioners do not yet apply screening methods; instead, they experiment with a few intuitively selected factors only. The following case study illustrates the need for screening. Bettonvil and Kleijnen (1996) present a greenhouse deterministic simulation model with 281 factors. The politicians wanted to take measures to reduce the release of CO_2 gasses; they realized that they should start with legislation for a limited number of factors. Another case study is presented by Kleijnen, Bettonvil, and Persson (2006), concerning a discrete-event simulation of a supply chain centered around an Ericsson company in Sweden. This simulation has 92 factors; the authors identify a shortlist with 10 factors after simulating only 19 combinations.

SB (like classic DOE) treats the simulation model as a *black box*; i.e., the simulation model transforms observable inputs into observable outputs, whereas the values of internal variables and specific functions implied by the simulation's computer modules are unobservable. The importance of factors depends on the *experimental domain*, so the users should supply information on this domain—including realistic ranges of the individual factors and limits on the admissible factor combinations; e.g., some factor values must add up to 100% in each combination.

SB uses the following metamodel *assumptions*.

1. A first-order polynomial augmented with two-factor interactions is a valid metamodel.
2. All first-order effects have known signs and are non-negative (so these effects cannot cancel each other out, when aggregated; see below).
3. There is 'strong heredity'; i.e., if a factor has no important main effect, then this factor does not interact with any other factor; also see Wu and Hamada (2000).

The SB procedure may be described roughly as follows. Its first step aggregates all factors into a single group, and runs the simulation model with that group at its low and high value respectively. SB compares these two simulation outputs; i.e. SB tests whether or not that group of factors has an important effect. If that group indeed has an important effect—which is most likely in the first step—then the second step splits the group into two subgroups— SB bifurcates. To test each of these subgroups for importance, SB runs the simulation with these subgroups as factors. In the next steps, SB splits important subgroups into smaller subgroups, and discards (freezes) unimportant subgroups. In the final step, all individual factors that are not in subgroups identified as unimportant, are estimated and tested.

This procedure may be interpreted though the following *metaphor*. Imagine a lake that is controlled by a dam. The goal of the experiment is to identify the highest (most important) rocks; actually, SB not only identifies but also measures the height of these 'rocks'. The dam is controlled in such a way that the level of the murky water slowly drops. Obviously, the highest rock first emerges from the water! The most-important-but-one rock turns up next, etc. SB stops when the analysts feel that all the 'important' factors are identified; once SB stops, the analysts know that all remaining (unidentified) factors have smaller effects than the effects of the factors that have been identified. This property of SB is important for its use in practice.

There is a need for more research:

- It is a challenge to derive the number of replicates that control the overall probability of correctly classifying the individual factors as important or unimportant. So far, SB applies a statistical test to each subgroup individually.
- After SB stops, the resulting shortlist of important factors should be validated.
- Multivariate (instead of univariate) output should be investigated.
- Software needs to be developed that implements SB.
- A contest may be organized for different screening methods and different simulation models. Such 'testbeds' are popular in MP.

4 KRIGING

This section reviews Kriging, and is based on Kleijnen (008a) and Kleijnen (008d). It presents the basic Kriging assumptions. This section also extends Kriging to random simulation, and discusses bootstrapping to estimate the variance of the Kriging predictor. Besides classic one-shot statistical designs such as Latin Hypercube Sampling (LHS), this section reviews sequentialized or customized designs for SA and optimization. It ends with topics for future research.

Typically, Kriging models are fitted to data that are obtained for larger experimental areas than the areas used in low-order polynomial regression; i.e., Kriging models are *global* (not local). Kriging is used for prediction (not explanation); its final goals are SA and optimization.

Kriging was originally developed in geostatistics—also known as spatial statistics—by the South African mining engineer Danie Krige. A classic geostatistics textbook is Cressie (1993). Later on, Kriging was applied to the I/O data of deterministic simulation models; see Sacks et al. (1989). Only recently Van Beers and Kleijnen (2003) applied Kriging to random simulation models. Ankenman, Nelson, and Staum (2008) present a detailed analysis of Kriging in random simulation. Although Kriging in random simulation is still rare, the track record of Kriging in deterministic simulation holds great promise for Kriging in random simulation.

This section focuses on the simplest type of Kriging called *Ordinary* Kriging, which assumes $w(\mathbf{d}) = \mu + \delta(\mathbf{d})$where $w(\mathbf{d})$ denotes the simulation output for input combination $\mathbf{d}$, μ is the simulation output averaged over the whole experimental area, and $\delta(\mathbf{d})$ is the additive noise that forms a *stationary covariance process* with zero mean.

Kriging uses the following *linear* predictor $y(\mathbf{d}) = \boldsymbol{\lambda}'\mathbf{w}$ where the weights $\boldsymbol{\lambda}$ are not constants—whereas the regression parameters (say) $\boldsymbol{\beta}$ are—but decrease with the *distance* between the 'new' input $\mathbf{d}$ to be predicted and the 'old' combinations, which are collected in the $n \times k$ design matrix $\mathbf{D}$.

The *optimal* weights can be proven to depend on $\boldsymbol{\Gamma} = (cov(w_i, w_{i'}))$ with $i, i' = 1, \ldots, n$ is the $n \times n$ matrix with the covariances between the 'old' outputs; $\boldsymbol{\gamma} = (cov(w_i, w_0))$ is the n-dimensional vector with the covariances between the n old outputs w_i and w_0, the output of the combination to be predicted; w_0 may be either new or old. These covariances are often based on the *correlation function*

$$\rho = \exp[-\sum_{j=1}^{k} \theta_j h_j^{p_j}] = \prod_{j=1}^{k} \exp[-\theta_j h_j^{p_j}] \qquad (1)$$

where h_j denotes the distance between the input d_j of the new and the old combinations, θ_j denotes the importance of input j (the higher θ_j is, the less effect input j has), and p_j denotes the smoothness of the correlation function (e.g., $p_j = 2$ implies an infinitely differentiable function). Exponential and Gaussian correlation functions have $p_j = 1$ and $p_j = 2$ respectively.

This correlation function implies that the weights are relatively high for inputs close to the input to be predicted. Furthermore, some of the weights may be negative. Finally, the weights imply that for an old input the predictor equals the observed simulation output at that input: $y(\mathbf{d}_i) = w(\mathbf{d}_i)$ with $\mathbf{d}_i \in \mathbf{D}$, so all weights are zero except the weight of the observed output, which is one; i.e., the Kriging predictor is an *exact interpolator*. Note that the OLS regression predictor minimizes the Sum of Squared Residuals (SSR), so it is not an exact interpolator—unless $n = q$ (saturated design).

A major problem is that the optimal weights in (**??**) depend on the correlation function of the underlying simulation model (e.g., (1))—*but this correlation function is unknown*. Therefore both the type and the parameter values must be estimated. To estimate the parameters of such a correlation function, the standard software and literature uses Maximum Likelihood Estimators (MLEs). The estimation of the correlation functions and the corresponding optimal weights in (**??**) can be done through DACE, which is software that is well documented and free of charge; see Lophaven, Nielsen, and Sondergaard (2002).

The interpolation property is attractive in *deterministic* simulation, because the observed simulation output is unambiguous. In *random* simulation, however, the observed output is only one of the many possible values. For random simulations, Van Beers and Kleijnen (2003) replaces $w(\mathbf{d}_i)$ by the average observed output $\overline{w_i}$. Those authors give examples in which the Kriging predictions are much better than the regression predictions (regression metamodels may be useful for other goals; e.g., understanding, screening, and V & V). Ankenman, Nelson, and Staum (2008) present a Kriging predictor that is no longer an interpolator in random simulation. Kleijnen (008a) also discusses Kriging in random simulation.

The literature virtually ignores problems caused by replacing the weights $\boldsymbol{\lambda}$ in (**??**) by the estimated optimal weights (say) $\widehat{\boldsymbol{\lambda}_0}$. Nevertheless, this replacement makes the Kriging predictor a *nonlinear* estimator. The literature uses the predictor variance—*given* the Kriging weights. This variance implies a zero variance in case the new point w_0 equals an old point w_i. Furthermore this equation tends to underestimate the true variance. Finally, this conditional variance and the true variance do not reach their maxima for the same input combination, which is important in sequential designs. See Den Hertog, Kleijnen, and Siem (2006) for details.

In random simulation, each input combination is replicated a number of times so a simple method for estimating the true predictor variance is *distribution-free bootstrapping*. To estimate the predictor variance, Van Beers and Kleijnen (2008) resample—with replacement—the (say) m_i replicates for combination i ($i = 1,\ldots,n$). This sampling results in the bootstrapped average $\overline{w_i^*}$ where the superscript $*$ is the usual symbol to denote a bootstrapped observation. From these n bootstrapped averages $\overline{w_i^*}$, the bootstrapped estimated optimal weights $\widehat{\boldsymbol{\lambda}_0^*}$ and the corresponding bootstrapped Kriging predictor y^* are computed. To decrease sampling effects, this whole procedure is repeated B times (e.g., $B = 100$), which gives y_b^* with $b = 1,\ldots,B$. The variance of the Kriging predictor is estimated from these y_b^*.

Another issue in Kriging is how to select the input combinations that result in the I/O simulation data to which the Kriging model is fitted. Simulation analysts often use *LHS* (LHS was not invented for Kriging but for risk analysis; see Kleijnen (008a)). LHS assumes that a valid metamodel is more complicated than a low-order polynomial, which is assumed by classic designs. LHS does not assume a specific metamodel. Instead, LHS focuses on the design space formed by the k-dimensional unit cube defined by the k standardized simulation inputs. LHS is one of the space filling types of design (other designs are discussed in (Kleijnen 008a) and (Kleijnen 008d)).

Instead of a one-shot space-filling design such as a LHS design, a *sequentialized* design may be used. In general, sequential statistical procedures are known to require fewer observations than fixed-sample (one-shot) procedures; see Park et al. (2002). Sequential designs imply that observations are analyzed—so the data generating process is better understood—before the next input combination is selected. This property implies that the design depends on the specific underlying process (simulation model); i.e., the design is customized (tailored or application-driven, not generic). Furthermore, such a design is attractive in simulation because computer experiments (unlike real-life experiments) proceed sequentially.

A sequential design for Kriging in SA is developed in Van Beers and Kleijnen (2008); it has the following steps.

1. Start with a *pilot* experiment, using some small generic space-filling design; e.g., a LHS design.
2. Fit a *Kriging* model to the I/O simulation data that are available at this step (in the first pass of this procedure, these I/O data are the data resulting from Step 1).
3. Consider (but do not yet simulate) a set of *candidate* input combinations that have not yet been simulated and that are selected through some space-filling design; select as the next combination to be actually simulated, the candidate combination that has the *highest predictor variance*.
4. Use the combination selected in Step 3 as the input combination to the (expensive) simulation model, and obtain the corresponding simulation output.
5. Return to Step 2, unless the Kriging metamodel is acceptable for its goal (SA).

The resulting design is indeed *customized*; i.e., which combination has the highest predictor variance is determined by the underlying simulation model; e.g., for the classic M/M/1 this design selects relatively few low traffic rates that give a less steep I/O function. A sequential design for constrained optimization (instead of SA) will be presented in Section 5.2.

I see a need for more research on Kriging in simulation:

- For random simulation, Kriging *software* needs further improvement; e.g., allow predictors that do not equal the average outputs at the input combinations already observed; see Ankenman et al. (2008) and Kleijnen (008a).
- Sequential designs may benefit from *asymptotic proofs* of their performance; e.g., does the design approximate the optimal design?
- More experimentation and analyses may be done to derive *rules of thumb* for the sequential design's

parameters, such as the size of the pilot design and the initial number of replicates.
- *Stopping rules* for sequential designs based on a measure of accuracy may be investigated.
- Nearly all Kriging publications assume univariate output, whereas in practice simulation models have *multivariate output*.
- Often the analysts know that the simulation's I/O function has certain properties, e.g., monotonicity. Most metamodels (such as Kriging and regression) do not preserve these properties.

5 OPTIMIZATION

The importance of the optimization of engineered systems is emphasized in a 2006 NSF panel; see Oden (2006). That report also points out the crucial role of simulation in engineering science. There are many methods for simulation optimization; see Kleijnen (008a) and the WSC proceedings. Section 5.1 reviews RSM; Section 5.2 reviews Kriging combined with MP; and Section 5.3 reviews robust simulation-optimization.

5.1 RSM

This subsection is based on Kleijnen (008e), which summarizes Generalized RSM (GRSM), extending Box and Wilson's RSM originally developed for real-life systems (that RSM is also covered in (Myers and Montgomery 1995)). GRSM allows multiple (multivariate) random responses, selecting one response as goal and the other responses as constrained variables. Both GRSM and RSM estimate local gradients to search for the optimum. These gradients are based on local first-order polynomial approximations. GRSM combines these gradients with MP findings to estimate a better search direction than the Steepest Descent (SD) direction used by RSM; see (2) below. Moreover, GRSM uses these gradients in a bootstrap procedure for testing the Karush-Kuhn-Tucker (KKT) conditions for the estimated optimum.

Classic RSM has the following *characteristics*.

- RSM is an *optimization heuristic* that tries to estimate the input combination that minimizes a given univariate goal function.
- RSM is a *stepwise* (multi-stage) method.
- In these steps, RSM uses local first-order and second-order *polynomial* metamodels (response surfaces). RSM assumes that these models have *white noise* in the local experimental area; when moving to a new local area in a next step, the variance may change.
- To fit these first-order polynomials, RSM uses *classic R-III designs*; for second-order polynomials, RSM usually applies a CCD.
- To determine in which direction the inputs will be changed in a next step, RSM uses SD based on the *estimated gradient* $\widehat{\boldsymbol{\beta}}_{-0} = (\widehat{\beta}_1, \ldots, \widehat{\beta}_k)'$ implied by the first-order polynomial fitted in the current step; the subscript -0 means that the intercept $\widehat{\beta}_0$ vanishes in the estimated gradient.
- In the final step, RSM takes the *derivatives* of the locally fitted second-order polynomial to estimate the optimum input combination. RSM may also apply *canonical analysis* to examine the shape of the optimal (sub)region: unique minimum, saddle point, ridge?

Kleijnen, den Hertog, and Angün (2006) derive *Adapted Steepest Descent* (ASD), which uses $[\mathbf{cov}(\widehat{\boldsymbol{\beta}}_{-\mathbf{0}})]^{-1}\widehat{\boldsymbol{\beta}}_{-0}$; e.g., the higher the variance of a factor effect is, the less the search moves into the direction of that factor. ASD gives a scale-independent search direction, and in general performs better than SD.

In practice, simulation models have *multiple outputs* so GRSM is more relevant than RSM. GRSM generalizes SD (applied in RSM) through ideas from *interior point* methods in MP. This search direction moves faster to the optimum than SD, since the GRSM avoids creeping along the boundary of the feasible area determined by the constraints on the random outputs and the deterministic inputs. GRSM's search direction is scale independent. More specifically, this *search direction* is

$$\mathbf{d} = -\left(\mathbf{B}'\mathbf{S}^{-2}\mathbf{B} + \mathbf{R}^{-2} + \mathbf{V}^{-2}\right)^{-1}\widehat{\boldsymbol{\beta}}_{0;-0} \qquad (2)$$

where $\mathbf{B}$ is the matrix with the gradients of the constrained outputs, $\mathbf{S}$, $\mathbf{R}$, and $\mathbf{V}$ are diagonal matrixes with the current estimated slack values for the constrained outputs, and the lower and upper limits for the deterministic inputs, and $\widehat{\boldsymbol{\beta}}_{0;-0}$ is the classic estimated SD direction for the goal output.

Analogously to RSM, GRSM proceeds *stepwise*; i.e., after each step along the search path (2), the following hypotheses are tested:

1. The simulated goal output of the new combination is *no improvement* over the old combination (pessimistic null-hypothesis).
2. This new combination is *feasible*; i.e., the other simulation outputs satisfy the constraints.

To test these hypotheses, we may apply the classic Student t statistic (a paired t statistic if CRN are used). Because multiple hypotheses are tested, Bonferroni's inequality may be used; i.e., divide the classic α value by the number of tests.

Actually, a better combination may lie in between the old and the new combinations. GRSM uses *binary search*; i.e., it simulates a combination that lies halfway these two combinations (and is still on the search path). This halving of the stepsize may be applied a number of times.

Next, GRSM proceeds analogously to RSM. So around the best combination found so far, it selects a new local area. Again a R-III design selects new simulation input combinations. And first-order polynomials are fitted for each type of simulation output, which gives a new search direction. And so on.

In random simulation the gradients and the slacks of the constraints must be estimated. This estimation turns the KKT first-order optimality conditions into a problem of nonlinear statistics. Angün and Kleijnen (2008) present an asymptotic test; Bettonvil, del Castillo, and Kleijnen (2008) derive a small-sample bootstrap test.

5.2 Kriging and MP

This subsection summarizes Kleijnen, van Beers, and van Nieuwenhuyse (2008), presenting a heuristic for constrained simulation-optimization (so it is an alternative for GRSM). There are additional constraints: the inputs must be *integers*, and they must satisfy non-box constraints The heuristic combines (i) sequential designs to specify the simulation inputs, (ii) Kriging metamodels to analyze the global I/O (whereas GRSM uses local metamodels), and (iii) Integer Non-Linear Programming (INLP) to estimate the optimal solution from the Kriging metamodels. The heuristic is applied to an (s, S) inventory system with a service (fill rate) constraint, and a realistic call-center simulation with a service constraint; the heuristic is compared with a popular commercial heuristic, namely OptQuest.

The heuristic is summarized in Figure 1. Some details are as follows.

1. The *pilot* design uses a standard maximin LHS design, which is a LHS design that maximizes the minimum distance between input points, and accounts for box constraints for the inputs. Moreover, the heuristic accounts for non-box input constraints; e.g., the sum of some inputs must meet a budget constraint.
2. The heuristic simulates all combinations of a design with the number of *replicates* depending on the signal/noise of the output.
3. To *validate* the Kriging metamodels, the heuristic applies *cross-validation*; see Kleijnen (008a).To estimate the variance of the Kriging predictor, the heuristic applies distribution-free bootstrapping to the replicates (accounting for a non-constant number of replicates per input combination, and CRN).
4. Some new combinations are selected to improve the fit of the *global* Kriging metamodel, whereas some other combinations are added because they seem to be close to the *local optimum*.

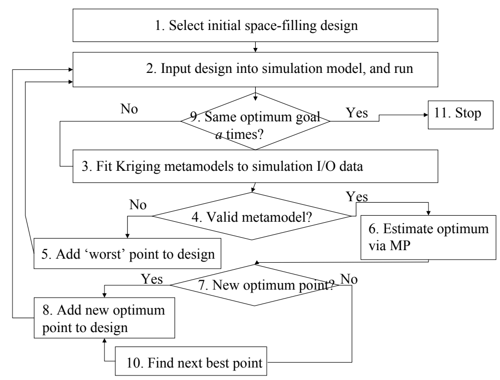

Figure 1: Kriging and MP flowchart.

5.3 Taguchian Robust Optimization

Whereas most simulation-optimization methods assume known environments, this subsection develops a 'robust' methodology for uncertain environments. This methodology uses Taguchi's view of the uncertain world, but replaces his statistical techniques by either RSM or Kriging combined with MP. Myers and Montgomery (1995) extend RSM to robust optimization of real-life systems. This subsection is based on Dellino, Kleijnen, and Meloni (2008), adapting robust RSM for simulated systems, including bootstrapping of the estimated Pareto frontier. Dellino et al. apply this method to a classic Economic Order Quantity (EOQ) inventory model, which demonstrates that a robust optimal order quantity may differ from the classic EOQ.

Taguchi originally developed his approach to help Toyota design 'robust' cars; i.e., cars that perform reasonably well in many circumstances (from the snows in Alaska to the sands in the Sahara); see Taguchi (1987) and Wu and Hamada (2000). Taguchi distinguishes between two types of variables:

- Decision (or control) factors (say) d_j $(j = 1, \ldots, k)$
- Environmental (or noise) factors, e_g $(g = 1, \ldots, c)$.

Taguchi assumes a single output (say) w. He focuses on the mean and the variance of this output.

Dellino et al. do not use Taguchi's statistical methods, because simulation enables the exploration of many more factors, factor levels, and factor combinations. Moreover, Taguchi uses a *scalar* output such as the signal-to-noise or mean-to-variance ratio; Dellino et al. allow each output to have a statistical distribution characterized through its mean and standard deviation; also see Myers and Montgomery (1995, p. 491). Dellino et al. solve the resulting bi-objective problem through the estimation of the *Pareto frontier*.

In the spirit of RSM, Myers and Montgomery (1995, p. 218, 492) assume:

- a *second-order* polynomial for the decision factors d_j;
- a first-order polynomial for the environmental factors e_g;
- *Control-by-noise two-factor interactions* (say) $\delta_{j;g}$,

resulting in

$$\begin{aligned} y &= \beta_0 + \sum_{j=1}^{k}\beta_j d_j + \sum_{j=1}^{k}\sum_{j'\geq j}^{k}\beta_{j;j'}d_j d_{j'} + \\ &+ \sum_{g=1}^{c}\gamma_j e_j + \sum_{j=1}^{k}\sum_{g=1}^{c}\delta_{j;g}d_j e_g + \varepsilon \\ &= \beta_0 + \boldsymbol{\beta}'\mathbf{d} + \mathbf{d}'\mathbf{B}\mathbf{d} + \boldsymbol{\gamma}'\mathbf{e} + \mathbf{d}'\boldsymbol{\Delta}\mathbf{e} + \varepsilon. \end{aligned} \tag{3}$$

Whereas Myers and Montgomery (1995, p. 493-494) assume that the environmental variables $\mathbf{e}$ satisfy $E(\mathbf{e}) = \mathbf{0}$ and $\mathbf{cov}(\mathbf{e}) = \sigma_e^2\mathbf{I}$, Dellino assume $E(\mathbf{e}) = \boldsymbol{\mu}_\mathbf{e}$ and $\mathbf{cov}(\mathbf{e}) = \boldsymbol{\Omega}_\mathbf{e}$ and derive from (3)

$$E(y) = \beta_0 + \boldsymbol{\beta}'\mathbf{d} + \mathbf{d}'\mathbf{B}\mathbf{d} + \boldsymbol{\gamma}'\boldsymbol{\mu}_\mathbf{e} + \mathbf{d}'\boldsymbol{\Delta}\boldsymbol{\mu}_\mathbf{e} \tag{4}$$

and

$$var(y) = (\boldsymbol{\gamma}' + \mathbf{d}'\boldsymbol{\Delta})\boldsymbol{\Omega}_\mathbf{e}(\boldsymbol{\gamma} + \boldsymbol{\Delta}'\mathbf{d}) + \sigma_\varepsilon^2 = \mathbf{l}'\boldsymbol{\Omega}_\mathbf{e}\mathbf{l} + \sigma_\varepsilon^2. \tag{5}$$

where $\mathbf{l} = (\boldsymbol{\gamma} + \boldsymbol{\Delta}'\mathbf{d}) = (\partial y/\partial e_1, \ldots, \partial y/\partial e_c)'$; i.e., $\mathbf{l}$ is the gradient with respect to the environmental factors—which follows directly from (3). So, the larger the gradient's components are, the larger the variance of the predicted simulation output is. Furthermore, if $\boldsymbol{\Delta} = \mathbf{0}$ (no control-by-noise interactions), then $var(y)$ cannot be controlled through the control variables $\mathbf{d}$.

Myers and Montgomery (1995, p. 495) discuss *constrained optimization*, which minimizes (e.g.) the variance subject to a constraint on the mean; see (4) and (5). They often simply superimpose contour plots for the mean and variance, to select an appropriate compromise or 'robust' solution. Dellino et al., however, use MP—which is more general and flexible.

To construct *confidence intervals* for the robust optimum, Myers and Montgomery (1995, p. 498) assume normality. Myers and Montgomery (1995, p. 504) notice that the analysis becomes complicated when the noise factors do not have constant variances. Dellino et al. therefore use *parametric bootstrapping*, which assumes that the distribution of the relevant random variable is known (in the EOQ example, the distribution is Gaussian).

OLS may be used to estimate the parameters in (4) and (5). The final goal of robust optimization is to minimize the resulting estimated mean $\widehat{y}$, while keeping the estimated standard deviation $\widehat{\sigma}_y$ below a given threshold. This constrained minimization problem may be solved through Matlab's 'fmincon', which gives the values of the 'estimated robust decision variables' (say) $\widehat{\mathbf{d}^+}$ and its corresponding mean $\widehat{y}$ and standard deviation $\widehat{\sigma}_y$. Next, varying the threshold value (say) 100 times may give up to 100 different solutions $\widehat{\mathbf{d}^+}$ with corresponding $\widehat{y}$ and $\widehat{\sigma}_y$. These 100 pairs $(\widehat{y}, \widehat{\sigma}_y)$ give the estimated Pareto frontier. To estimate the variability of this frontier, bootstrapping may be used.

Dellino et al. demonstrate robust optimization through an EOQ simulation, which is deterministic. They copy the EOQ parameter values from Hillier and Lieberman (2001), pp. 936–937, 942–943.

Dellino et al. assume that the demand per time unit is constant, but this constant (say) a is unknown. More specifically, a has a Gaussian distribution with mean μ_a and standard deviation σ_a where μ_a is the 'base' or 'nominal' value (used in the RSM optimization of the EOQ model) and σ_a quantifies the uncertainty about the true input parameter. Myers and Montgomery (1995, pp. 463–534) use only two values per environmental factor, which suffices to estimate its main effect and its interactions with the decision factors. Dellino et al., however, use LHS to select five values for the environmental factor a, because LHS is popular in risk analysis. These values are crossed with five values for the decision variable Q, as is usual in a Taguchian approach; Q has five values if a CCD is used. (Dellino et al. and Myers and Montgomery (1995, p. 487) discuss designs more efficient than crossed designs.)

The 'estimated robust optimal' order quantity (say) $\widehat{Q^+}$ is the quantity that minimizes the estimated mean cost $\widehat{C}$ while keeping the estimated standard deviation $\widehat{\sigma_C}$ below a given threshold T. This constrained minimization problem is solved through Matlab's fmincon. For example, $T = 42500$ gives $\widehat{Q^+} = 28568$, but $T = 41500$ gives $\widehat{Q^+} = 35222$; the classic EOQ is $\widehat{Q_o} = 28636$ so the difference between the two order quantities is nearly 25% if the managers are risk-averse (low threshold T). Because management cannot give a single, fixed value for the threshold, the threshold is varied—which gives the estimated *Pareto frontier*. This frontier demonstrates that if management prefers low costs variability, then they must pay a price; i.e., the expected cost increases; also see Figure 2.

Future research may address the following issues.

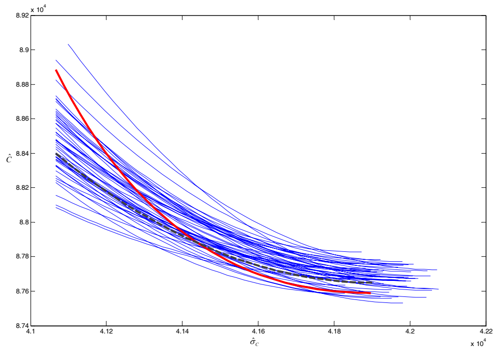

Figure 2: Bootstrapped Pareto frontiers.

- A better type of metamodel may be a *Kriging* model.
- The methodology needs adjustment for *random* simulation models, with scalar output or vector output.
- *Integer* constraints on some input variables may be needed.

6 CONCLUSIONS

DOE for random simulation borrows many techniques from DOE for real-life experiments—such as R-III designs and CCDs with low-order polynomial metamodels—and deterministic simulation—such as LHS and Kriging. Random simulation, however, uses PRN (and CRN) so it involves (possibly correlated) internal noise—besides errors caused by lack-of-fit. Simulation allows experimentation with many factors, so screening designs such as SB are very important and deserve more research and application. Kriging is already popular in deterministic simulation, but Kriging in random simulation deserves additional research and more applications. Moreover, Kriging may be combined with sequential designs, for either SA or optimization. Most optimization literature has focused on simulation with a single output, whereas practical simulation generates multiple outputs. These multiple outputs may be handled by methods such as GRSM and MP combined with Kriging—which also deserve more research and application. Finally, robust optimization in random simulation has only started—even though it is a very important topic in practice. For all these different types of design and analysis, it is urgent to develop software that is easily combined with software for random simulation modeling and analysis; also see Schruben (2008).

REFERENCES

Angün, E., and J. P. C. Kleijnen. 2008. An asymptotic test of optimality conditions in multiresponse simulation-based optimization. Working paper.

Ankenman, B., B. L. Nelson, and J. Staum. 2008. Stochastic kriging for simulation metamodeling. In *Proceedings of the 2008 Winter Simulation Conference*.

Bettonvil, B. 1990. *Detection of important factors by sequential bifurcation*. Ph.d. dissertation, Tilburg University Press, Tilburg.

Bettonvil, B., E. del Castillo, and J. P. C. Kleijnen. 2008. Statistical testing of optimality conditions in multiresponse simulation-based optimization. Working paper.

Bettonvil, B., and J. P. C. Kleijnen. 1996. Searching for important factors in simulation models with many factors: sequential bifurcation. *European Journal of Operational Research* 96 (1): 180–194.

Cressie, N. A. C. 1993. *Statistics for spatial data: revised edition*. New York: Wiley.

Dellino, G., J. P. C. Kleijnen, and C. Meloni. 2008. Robust optimization in simulation: Taguchi and response surface methodology. Working paper.

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* 57 (4): 400–409.

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

Hillier, F. S., and G. J. Lieberman. 2001. *Introduction to operations research; seventh edition*. Boston: McGraw Hill.

Kleijnen, J. P. C. 2008a. *Design and analysis of simulation experiments*. Springer.

Kleijnen, J. P. C. 2008b. Simulation experiments in practice: statistical design and regression analysis. *Journal of Simulation* 2 (1): 19–27.

Kleijnen, J. P. C. 2008c. Factor screening in simulation experiments: review of sequential bifurcation. In *Advancing the Frontiers of Simulation: A Festschrift in Honor of George S. Fishman*, ed. C. Alexopoulos, D. Goldsman, and J. R. Wilson. New York: Springer. In preparation.

Kleijnen, J. P. C. 2008d. Kriging metamodeling in simulation: a review. *European Journal of Operational Research*. Accepted.

Kleijnen, J. P. C. 2008e. Response surface methodology for constrained simulation optimization: an overview. *Simulation Modelling Practice and Theory* 16:50–64.

Kleijnen, J. P. C., B. Bettonvil, and F. Persson. 2006. Screening for the important factors in large discrete-event simulation: sequential bifurcation and its applications. *Screening: Methods for experimentation in industry, drug discovery, and genetics*:287–307.

Kleijnen, J. P. C., D. den Hertog, and E. Angün. 2006. Response surface methodology's steepest ascent and step size revisited: correction. *European Journal of Operational Research* 170:664–666.

Kleijnen, J. P. C., S. M. Sanchez, T. W. Lucas, and T. M. Cioppa. 2005. State-of-the-art review: a user's guide to the brave new world of designing simulation experiments. *INFORMS Journal on Computing* 17 (3): 263–289.

Kleijnen, J. P. C., and R. G. Sargent. 2000. A methodology for the fitting and validation of metamodels in simulation. *European Journal of Operational Research* 120 (1): 14–29.

Kleijnen, J. P. C., W. C. M. van Beers, and I. van Nieuwenhuyse. 2008. Constrained optimization in simulation: novel approach. Working paper.

Koehler, J. R., and A. B. Owen. 1996. Computer experiments. In *Handbook of statistics*, ed. S. Ghosh and C. Rao, Volume 13, 261–308. Amsterdam: Elsevier.

Lophaven, S. N., H. B. Nielsen, and J. Sondergaard. 2002. Dace: a matlab kriging toolbox, version 2.0. Technical report, IMM Technical University of Denmark, Lyngby.

Montgomery, D. C. 2009. *Design and analysis of experiments; 7th edition*. Hoboken, NJ: Wiley.

Myers, R. H., and D. C. Montgomery. 1995. *Response surface methodology: Process and product optimization using design experiments*. New York: Wiley.

Oden, J. T., C. 2006. Revolutionizing engineering science through simulation. Blue ribbon panel on simulation-based engineering science, National Science Foundation (NSF).

Park, S., J. W. Fowler, G. T. Mackulak, J. B. Keats, and W. M. Carlyle. 2002. D-optimal sequential experiments for generating a simulation-based cycle time-throughput curve. *Operations Research* 50 (6): 981–990.

Sacks, J., W. J. Welch, T. J. Mitchell, and H. P. Wynn. 1989. Design and analysis of computer experiments (includes comments and rejoinder). *Statistical Science* 4 (4): 409–435.

Santner, T. J., B. J. Williams, and W. I. Notz. 2003. *The design and analysis of computer experiments*. New York: Springer-Verlag.

Schruben, L. W. 2008. Simulation modeling for analysis. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*. Accepted.

Schruben, L. W., and B. H. Margolin. 1978. Pseudorandom number assignment in statistically designed simulation and distribution sampling experiments. *Journal American Statistical Association* 73 (363): 504–525.

Taguchi, G. 1987. *System of experimental designs, volumes 1 and 2*. New York: UNIPUB/ Krauss International, White Plains.

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.

Van Beers, W. C. M., and J. P. C. Kleijnen. 2008. Customized sequential designs for random simulation experiments: Kriging metamodeling and bootstrapping. *European Journal of Operational Research* 186 (3): 1099–1113.

Wu, C. F. J., and M. Hamada. 2000. *Experiments: planning, analysis, and parameter design optimization*. New York: Wiley.

Xu, J., F. Yang, and H. Wan. 2007. Controlled sequential bifurcation for software reliability study. In *Proceedings of the 2007 Winter Simulation Conference*, ed. S. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. Tew, and R. Barton, 281–288.

Yu, H.-F. 2007. Designing a screening experiment with a reciprocal weibull degradation rate. *Computers & Industrial Engineering* 52 (2): 175–191.

AUTHOR BIOGRAPHY

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 CentER. He also teaches at the Eindhoven University of Technology, namely the course 'Simulation for logistics' for the Postgraduate International Program in Logistics Management Systems. His research concerns simulation (especially its statistical design and analysis), information systems, and supply chains. He has been a consultant for several organizations in the USA and Europe. He serves on many international editorial boards and scientific committees. He spent some years in the USA, at universities and private companies. He received a number of national and international awards. More details are given on his website <http://center.uvt.nl/staff/kleijnen/>.