CentER

Discussion Paper

No. 2004–17

# DESIGN AND ANALYSIS OF MONTE CARLO EXPERIMENTS

By J.P.C. Kleijnen

February 2004

ISSN 0924-7815

TILBURG UNIVERSITY

CS_Handbook_version_2.doc; written 2 February 2004; printed 2/2/2004 3:44 PM

# Design and analysis of Monte Carlo experiments

**Jack P.C. Kleijnen**

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

Chapter III. 3 of
Handbook of computational statistics
**Volume I: concepts and fundamentals**

Edited by
James E. Gentle <jgentle@gmu.edu> George Mason University
Wolfgang Haerdle <haerdle@wiwi.hu-berlin.de> Humboldt-Universitat zu Berlin
Yuichi Mori <mori@soci.ous.ac.jp> Okayama University of Science

To be published by
Springer-Verlag, Heidelberg, Germany

## 1. Introduction

By definition, computer simulation (or Monte Carlo) models are not solved by mathematical analysis (for example, differential calculus), but are used for numerical experimentation. These experiments are meant to answer questions of interest about the real world; i.e., the experimenters may use their simulation model to answer *what if* questions—this is also called *sensitivity analysis*. Sensitivity analysis—guided by the statistical theory on *design of experiments* (DOE)—is the focus of this chapter. Sensitivity analysis may further serve validation, optimization, and risk (or uncertainty) analysis for finding robust solutions; see Kleijnen (1998), Kleijnen et al. (2003a) and Kleijnen et al. (2003b). Note that optimization is also discussed at length in Chapter II.6 by Spall.

Though I assume that the reader is familiar with basic simulation, I shall summarize a simple Monte Carlo example (based on the well-known Student *t* statistic) in Section 2. This example further illustrates bootstrap and variance reduction techniques

Further, I assume that the reader 's familiarity with DOE is restricted to elementary DOE. In this chapter, I summarize classic DOE, and extend it to newer methods (for example, DOE for interpolation using Kriging; Kriging is named after the South-African mining engineer D.G. Krige).

Traditionally, 'the shoemaker's children go barefoot'; i.e., users of computational statistics ignore statistical issues—such as sensitivity analysis—of their simulation results. Nevertheless, they should address *tactical* issues—the number of (macro)replicates, variance reduction techniques—and *strategic* issues—situations to be simulated and the sensitivity analysis of the resulting data. Both types of issues are addressed in this chapter.

Note the following terminology. DOE speaks of 'factors' with 'levels' whereas simulation analysts may speak of 'inputs' or 'parameters' with 'values'. DOE talks about 'design points' or 'runs', whereas simulationists may talk about 'situations', 'cases', or 'scenarios'.

Classic DOE methods for real, non-simulated systems were developed for agricultural experiments in the 1930s, and—since the 1950s—for experiments in engineering, psychology, etc. (Classic designs include fractional factorials, as we shall

see.) In those real systems it is impractical to experiment with ‘many’ factors; $k = 10$ factors seems a maximum. Moreover, it is then hard to experiment with factors that have more than ‘a few’ values; five values per factor seems a maximum. Finally, these experiments are run in ‘one shot’—for example, in one growing season— and not sequentially. In simulation, however, these limitations do not hold!

Two textbooks on classic DOE for simulation are Kleijnen (1975, 1987). An update is Kleijnen (1998). A bird-eye’s view of DOE in simulation is Kleijnen et al. (2003a), which covers a wider area than this review.

Note further the following terminology. I speak of the *Monte Carlo* method whenever (pseudo)random numbers are used; for example, I apply the Monte Carlo method to estimate the behavior of the *t* statistic in case of non-normality, in Section 2 (the Monte Carlo method may also be used to estimate multiple integrals, which is a deterministic problem, outside the scope of this handbook). I use the term *simulation* whenever the analysts compute the output of a dynamic model; i.e., the analysts do not use calculus to find the solution of a set of differential or difference equations. The dynamic model may be either stochastic or deterministic. Stochastic simulation uses the Monte Carlo method; it is often applied in telecommunications and logistics. Deterministic simulation is often applied in computer-aided engineering (CAE). Finally, I use the term *metamodel* for models that approximate—or model—the input/output (I/O) behavior of the underlying simulation model; for example, a polynomial regression model is a popular metamodel (as we shall see). Metamodels are used—consciously or not—to design and analyze experiments with simulation models. In the simulation literature, metamodels are also called response surfaces, emulators, etc.

The remainder of this chapter is organized as follows. Section 2 presents a simple Monte Carlo experiment with Student’s *t* statistic, including bootstrapping and variance reduction techniques. Section 3 discusses the black box approach to simulation experiments, and corresponding metamodels——especially, polynomial and Kriging models. Section 4 starts with simple regression models with a single factor; proceeds with designs for multiple factors including designs for first-order and second-order polynomial models, and concludes with screening designs for hundreds of factors. Section 5 introduces Kriging interpolation, which has hardly been applied in random simulation—but has already established a track record in deterministic simulation and spatial statistics. Kriging often uses space-filling designs, such as

Latin hypercube sampling (LHS). Section 6 gives conclusions and further research topics.

## 2. Simulation techniques in computational statistics

Consider the well-known definition of the *t* statistic with $n - 1$ degrees of freedom:

$$t_{n-1} = \frac{\bar{x} - \mu}{s_x / \sqrt{n}} \tag{1}$$

where the $x_i$ ($i = 1, \ldots, n$) are assumed to be normally (Gaussian), independently, and identically distributed (NIID) with mean $\mu$ and variance $\sigma^2$:

$$x_i \in NIID(\mu, \sigma) \; (i = 1, \ldots, n) \tag{2}$$

Nearly 100 years ago, Gossett used a kind of Monte Carlo experiment (without using computers, since they were not yet invented), before he analytically derived the density function of this statistic (and published his results under the pseudonym of Student). So, he sampled *n* values $x_i$ (from an urn) satisfying (2), and computed the corresponding value for the statistic defined by (1). This experiment he repeated (say) *m* times, so that he could compute the estimated density function (EDF)—also called the empirical cumulative distribution function (ECDF)— of the statistic. (Inspired by these empirical results, he did his famous analysis.)

Let us imitate his experiment, in the following simulation experiment (this procedure is certainly not the most efficient computer program).

i. Read the simulation inputs: $\mu$ (mean), $\sigma^2$ (variance), *n* (sample size), *m* (number of macro-replicates, used in step iv).

ii. Take *n* samples $x_i \in \text{NIID}(\mu, \sigma)$ (see equation 2 and Chapter II.2 by L'Ecuyer).

iii. Compute the statistic $t_{n-1}$ (see equation 1).

iv. Repeat steps ii and iii *m* times.

v. Sort the *m* values of $t_{n-1}$.

vi. Compute the EDF from the results in step v.

To verify this simulation program, we may compare the result (namely the EDF) with the results that are tabulated for Student's density function; for example, does our EDF give a 10% quantile that is not significantly different from the tabulated value (say) $t_{n-1;\,0.90}$. Next we may proceed to the following more interesting application.

We may drop the classic assumption formulated in (2), and experiment with *non-normal* distributions. It is easy to sample from such distributions (see again Chapter II.2). However, we are now confronted with several so-called *strategic* choices (also see Step i above): Which type of distribution should be selected (lognormal, exponential, etc.); which parameter values for that distribution type (mean and variance for the lognormal, etc.), which sample size (for asymptotic, 'large' *n*, the *t* distribution is known to be a good approximation for our EDF).

Besides these choices, we must face some *tactical* issues: Which number of macro-replicates *m* gives a good EDF; can we use special *variance reducing techniques* (VRTs)—such as common random numbers and importance sampling—to reduce the variability of the EDF? We explain these techniques briefly, as follows.

*Common random numbers* (CRN) mean that the analysts use the same (pseudo)random numbers (PRN)—symbol *r*— when estimating the effects of different strategic choices. For example, CRN are used when comparing the estimated quantiles $\hat{t}_{n-1;\,0.90}$ for various distribution types. Obviously, CRN reduces the variance of estimated differences, provided CRN creates positive correlations between the estimators $\hat{t}_{n-1;\,0.90}$ being compared.

*Antithetic variates* (AV) mean that the analysts use the complements (1 - *r*) of the PRN (*r*) in two 'companion' macro-replicates. Obviously, AV reduces the variance of the estimator averaged over these two replicates, provided AV creates negative correlation between the two estimators resulting from the two replicates.

*Importance sampling* (IS) is used when the analysts wish to estimate a *rare* event, such as the probability of the Student statistic exceeding the 99.999% quantile. IS increases that probability (for example, by sampling from a distribution with a fatter tail)—and later on, IS corrects for this distortion of the input distribution (through the likelihood ratio). IS is not so simple as CRN and AV—but without IS too much computer time may be needed. See Glasserman et al. (2000).

There are many more VRTs. Both CRN and AV are intuitively attractive and easy to implement, but the most popular one is CRN. The most useful VRT may be IS. In practice, the other VRTs often do not reduce the variance drastically so many users prefer to spend more computer time instead of applying VRTs. (VRTs are a great topic for doctoral research!) For more details on VRTs, I refer to Kleijnen and Rubinstein (2001).

Finally, the density function of the sample data $\boldsymbol{x}_i$ may not be an academic problem: Suppose a very limited set of historical data is given, and we must analyze these data while we know that these data do not satisfy the classic assumption formulated in (2). Then *bootstrapping* may help, as follows (also remember the six steps above).

i. Read the bootstrap sample size *B* (usual symbol in bootstrapping, comparable with *m*—number of macro-replicates—in step i above).

ii. Take *n* samples with replacement from the original sample $\boldsymbol{x}_i$; this sampling gives $\boldsymbol{x}_i^*$ (the superscript * denotes bootstrapped values, to be distinguished from the original values).

iii. From these $\boldsymbol{x}_i^*$ compute the statistic $\boldsymbol{t}_{n-1}^*$ (see equation 1).

iv. Repeat steps ii and iii *B* times.

v. Sort the *B* values of $\boldsymbol{t}_{n-1}^*$.

vi. Compute the EDF from the results in step v.

In summary, bootstrapping is just a Monte Carlo experiment—using resampling with replacement of a given data set. (There is also a parametric bootstrap, which comes even closer to our simulation of Gossett's original experiment.) Bootstrapping is further discussed in Efron and Tibshirani (1993) and in Chapter III.2 (by Mammen).

## 3. Black-box metamodels of simulation models

DOE treats the simulation model as a *black box*; i.e., only the inputs and outputs are observed and analyzed. For example, in the simulation of the *t* statistic (in Section 2) the simulation inputs (listed in Step i) are $\mu$ (mean), $\sigma^2$ (variance), *n* (sample size), and *m* (number of macro-replicates); this *m* is probably a tactical factor that is not of

interest to the user. Suppose the user is interested in the 90% quantile of the distribution function of the statistic in case of nonnormality. A *black box* representation of this example is:

$$t_{n-1;\,0.90} = t(\mu, \sigma, n, r_0) \tag{3}$$

where $t(.)$ denotes the mathematical function implicitly defined by the simulation program (outlined in steps i through vi in Section 2); $\mu$ and $\sigma$ now denote the parameters of the nonnormal distribution of the input $\boldsymbol{x}_i$ (for example, $\mu$ denotes how many exponential distributions with parameter $\sigma = \lambda$ are summed to form an Erlang distribution); $\boldsymbol{r}_0$ denotes the seed of the pseudorandom numbers.

One possible *metamodel* of the black box model in (3) is a Taylor series approximation—cut off after the first-order effects of the three factors, $\mu, \sigma, n$:

$$y = \beta_0 + \beta_1 \mu + \beta_2 \sigma + \beta_3 n + e \tag{4}$$

where $y$ is the metamodel predictor of the simulation output $\boldsymbol{t}_{\boldsymbol{n}-1;\,0.90}$ in (3); $\boldsymbol{\beta}^T = (\beta_0, \beta_1, \beta_2, \beta_3)$ denotes the parameters of the metamodel in (4), and $e$ is the noise—which includes both *lack of fit* of the metamodel and *intrinsic noise* caused by the pseudorandom numbers.

Besides the metamodel specified in (4), there are many alternative metamodels. For example, taking the logarithm of the inputs and outputs in (4) makes the first-order polynomial approximate relative changes; i.e., the parameters $\beta_1$, $\beta_2$, and $\beta_3$ become elasticity coefficients.

There are many—more complex—types of metamodels. Examples are Kriging models, neural nets, radial basis functions, splines, support vector regression, and wavelets; see the various chapters in Part III—especially Chapters III.5 (by Loader), III.7 (Müller), III.8 (Cizek), and III.15 (Laskov and Müller)—and also Clarke, Griebsch, and Simpson (2003) and Antioniadis and Pham (1998). I, however, will focus on two types that have established a track record in simulation:

- linear regression models (see Section 4)

- Kriging (see Section 5).

To estimate the parameters of whatever metamodel, the analysts must *experiment* with the simulation model; i.e., they must change the inputs (or factors) of the simulation, run the simulation, and analyze the resulting input/output data. This experimentation is the topic of the next sections.

## 4. Designs for linear regression models

### *4.1 Simple regression models for simulations with a single factor*

I start with the simplest metamodel, namely a first-order polynomial with a single factor. An example is the 'Student' simulation in Section 2, where I now assume that we are interested only in the power so $y$ in (4) now denotes the type II error predicted through the regression model. I further assume a single factor (say) $x = \sigma/n$ ('relative' variability; i.e., absolute variability corrected for sample size); see (4). Elementary mathematics proves that—to fit a straight line—it suffices to have two input/output observations; see 'local area 1' in Figure 1. It is simple to prove that the 'best' estimators of the regression parameters in (4) result if those two values are as far apart as 'possible'.

**INSERT Figure 1**

In practice, the analysts do not know over which *experimental area* a first-order polynomial is a 'valid' model. This validity depends on the goals of the simulation study; see Kleijnen and Sargent (2000).

So the analysts may start with a *local* area, and simulate the two (locally) extreme input values. Let's denote these two extreme values of the 'coded' variable $x$ by -1 and +1, which implies the following *standardization* of the original variable $z$:

$$x = \frac{z - \bar{z}}{(z_{\max} - z_{\min})/2} \tag{5}$$

where $\bar{z}$ denotes the average value of the relative variability $z = \sigma/n$ in the (local) experiment.

The Taylor series argument implies that—as the experimental area gets bigger (see 'local area 2' in Figure 1)—a better metamodel may be a second-order polynomial:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + e. \quad (6)$$

Obviously, estimation of the three parameters in (6) requires at least the simulation of three input values. Indeed, DOE provides designs with three values per factor; for example, $3^k$ designs. However, most publications on the application of DOE in simulation discuss *Central Composite Designs* (CCD), which have five values per factor; see Kleijnen (1975).

I emphasize that the second-order polynomial in (6) is nonlinear in $\boldsymbol{x}$ (the regression variable), but *linear* in $\boldsymbol{\beta}$ (the regression parameters or factor effects to be estimated ). Consequently, such a polynomial is a type of *linear regression* model (also see Chapter III.8).

Finally, when the experimental area covers the *whole* area in which the simulation model is valid (see again Figure 1), then other *global* metamodels become relevant. For example, Kleijnen and Van Beers (2003a) find that *Kriging* (discussed in Section 5) outperforms second-order polynomial fitting.

Note that Zeigler, Praehofer, and Kim (2000) call the experimental area the 'experimental frame'. I call it the domain of admissible scenarios, given the goals of the simulation study.

I conclude that *lessons* learned from the simple example in Figure 1, are:

i. The analysts should decide whether they want to experiment *locally or globally*.

ii. Given that decision, they should select a specific *metamodel type* (low-order polynomial, Kriging, spline, etc.); also see Chapters III.5, III.7, and III.8.

*4.2 Simple regression models for simulation models with multiple factors*

Let's now consider a regression model with *k* factors; for example, *k* = 2. The design that is still most popular—even though it is inferior— *changes one factor at a time*. For *k* = 2 such a design is shown in Figure 2 and Table 1; in this table the factor values over the various factor combinations are shown in the columns denoted by $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$; the 'dummy' column $\boldsymbol{x}_0$ corresponds with the polynomial intercept $\hat{\beta}_0$ in (4). In this design the analysts usually start with the 'base' scenario, denoted by the factor combination (0, 0); see scenario 1 in the table. Next they run the two scenarios (1, 0) and (0, 1); see the scenarios 2 and 3 in the table..

In a one-factor-at-a-time design, the analysts cannot estimate the *interaction* between the two factors. Indeed, Table 1 shows that the estimated interaction (say) $\beta_{1;2}$ is *confounded* with the estimated intercept $\hat{\beta}_0$; i.e., the columns for the corresponding regression variables are linearly dependent. (Confounding remains when the base values are denoted not by zero but by one; then these two columns become identical.)

**INSERT Figure 2**

**INSERT Table 1**

In practice, analysts often study each factor at *three levels* (which may be denoted by -1, 0, +1) in their one-at-a-time design. However, two levels suffice to estimate the parameters of a first-order polynomial (see again Section 4.1).

To enable the estimation of interactions, the analysts must change factors *simultaneously*. An interesting problem arises if *k* increases from two to three. Then Figure 2 becomes Figure 3, which does not show the output (*w*), since it would require a fourth dimension (instead $x_3$ replaces *w*); the asterisks are explained in Section 4.3. And Table 1 becomes Table 2. The latter table shows the $2^3$ factorial design; i.e., in the experiment each of the three factors has two values and all their combinations of values are simulated. To simplify the notation, the table shows only the signs of the factor values, so - means -1 and + means +1. The table further shows possible regression variables, using the symbols '0' through '1.2.3'—to denote the indexes of the regression variables $\boldsymbol{x}_0$ (the dummy, always equal to 1) through

$\boldsymbol{x}_1\boldsymbol{x}_2\boldsymbol{x}_3$ (third-order interaction). Further, I point out that each column is *balanced*; i.e., each column has four plusses and four minuses —except for the dummy column.

**INSERT Table 2**

The $2^3$ design enables the estimation of all eight parameters of the following regression model, which is a third-order polynomial that is *incomplete*; i.e., some parameters are assumed zero:

$$y = \beta_0 + \sum_{j=1}^{3} \beta_j x_j + \sum_{j=1}^{2} \sum_{j'>j}^{3} \beta_{j;j'} x_j x_{j'} + \beta_{1;2;3} x_1 x_2 x_3 + e\,. \tag{7}$$

**INSERT Figure 3**

Indeed, the $2^3$ design implies a matrix of regression variables $\boldsymbol{X}$ that is *orthogonal*:

$$\boldsymbol{X}^T\boldsymbol{X} = n\boldsymbol{I} \tag{8}$$

where $n$ denotes the number of scenarios simulated; $n$ = 8 in Table 2. Hence the *ordinary least squares* (OLS) estimator

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{w} \tag{9}$$

simplifies for the $2^3$ design —which is orthogonal so (8) holds—to $\hat{\boldsymbol{\beta}} = \boldsymbol{X}^T\boldsymbol{w}/8$.

The *covariance matrix* of the (linear) OLS estimator given by (9) is

$$\boldsymbol{cov}(\hat{\boldsymbol{\beta}}) = [(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T]\boldsymbol{cov}(\boldsymbol{w})\,[(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T]^T\,. \tag{10}$$

In case of *white noise*; i.e.,

$$\boldsymbol{cov}(\boldsymbol{w}) = \sigma^2\boldsymbol{I}\,, \tag{11}$$

(10) reduces to the well-known formula

$$\boldsymbol{cov}(\hat{\boldsymbol{\beta}}) = \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}. \quad (12)$$

However, I claim that in practice this white noise assumption does not hold:

i. The output variances change as the input changes so the assumed common variance $\sigma^2$ in (11) does not hold. This is called *variance heterogeneity*. (Well-known examples are Monte Carlo studies of the type I and type II errors, which are binomial variables so the estimated variances are $y(1 - y)/m$; also see Section 2)

ii. Often the analysts use *common random numbers* (see CRN in Section 2), so the assumed diagonality of the matrix in (11) does not hold.

Therefore I conclude that the analysts should choose between the following two options.

i. Continue to apply the OLS *point* estimator (9), but use the *covariance* formula (10) instead of (12)

ii. Switch from OLS to *Generalized Least Squares* (GLS) with $\boldsymbol{cov}(\boldsymbol{w})$ estimated from $m > n$ replications (so the estimated covariance matrix is not singular); for details see Kleijnen (1992, 1998).

The variances of the estimated regression parameters—which are on the main diagonal of $\boldsymbol{cov}(\hat{\boldsymbol{\beta}})$ in (10)—can be used to test statistically whether some factors have zero effects. However, I emphasize that a significant factor may be unimportant—practically speaking. If the factors are scaled between –1 and +1 (see the transformation in (5)), then the estimated effects quantify the *order of importance*. For example, in a first-order polynomial regression model the factor estimated to be the most important factor is the one with the highest absolute value for its estimated effect. See Bettonvil and Kleijnen (1990).

*4.3 Fractional factorial designs and other incomplete designs*

The incomplete third-order polynomial in (7) included a third-order effect, namely $\beta_{1;2;3}$. Standard DOE textbooks include the definition and estimation of such high-order interactions. However, the following claims may be made:

i. High-order effects are hard to interpret

ii. These effects often have negligible magnitudes.

Claim # 1 seems obvious. If claim #2 holds, then the analysts may simulate fewer scenarios than specified by a full factorial (such as the $2^3$ design). For example, if $\beta_{1;2;3}$ is indeed zero, then a $2^{3-1}$ fractional factorial design suffices. A possible $2^{3-1}$ design is shown in Table 2, deleting the four rows (scenarios) that have a minus sign in the 1.2.3 column (i.e., delete the rows 1, 4, 6, 7). In other words, only a *fraction*—namely $2^{-1}$ of the $2^3$ full factorial design—is simulated. This design corresponds with the points denoted by the symbol * in Figure 3. Note that this figure has the following geometrically property: each scenario corresponds with a vertex that cannot be reached via a single edge of the cube.

In this $2^{3-1}$ design two columns are identical, namely the 1.2.3 column (with four plusses) and the dummy column. Hence, the corresponding two effects are confounded—but the high-order interaction $\beta_{1;2;3}$ is assumed zero, so this confounding can be ignored!

Sometimes a *first-order polynomial* suffices. For example, in the (sequential) optimization of black-box simulation models the analysts may use a first-order polynomial to estimate the local gradient; see Angün et al. (2002). Then it suffices to take a $2^{k-p}$ design with the biggest $p$ value that makes the following condition hold: $2^{k-p} > k$. An example is: $k = 7$ and $p = 4$ so only 8 scenarios are simulated; see Table 3. This table shows that the first three factors (labeled 1, 2, and 3) form a full factorial $2^3$ design; the symbol '4 = 1.2' means that the values for factor 4 are selected by multiplying the elements of the columns for the factors 1 and 2. Note that the design is still balanced and orthogonal. Because of this orthogonality, it can be proven that the estimators of the regression parameters have smaller variances than one-factor-at-a-time designs give. How to select scenarios in $2^{k-p}$ designs is discussed in many DOE textbooks, including Kleijnen (1975, 1987).

**INSERT Table 3**

Actually, these designs—i.e., fractional factorial designs of the $2^{k-p}$ type with biggest $p$ value still enabling the estimation of first-order polynomial regression models—are a subset of *Plackett-Burman designs*. The latter designs consists of $k + 1$ combinations with $k + 1$ rounded upwards to a multiple of four; for example, if $k = 11$,

then Table 4 applies. If $k = 8$, then the Plackett-Burman design is a $2^{7-4}$ fractional factorial design; see Kleijnen (1975, pp. 330-331). Plackett-Burman designs are tabulated in many DOE textbooks, including Kleijnen (1975). Note that designs for first-order polynomial regression models are called *resolution III* designs.

**INSERT Table 4**

*Resolution IV* designs enable unbiased estimators of first-order effects—even if two-factors interactions are important. These designs require double the number of scenarios required by resolution III designs; i.e., after simulating the scenarios of the resolution III design, the analysts simulate the *mirror scenarios*; i.e., multiply by –1 the factor values in the original scenarios.

*Resolution V* designs enable unbiased estimators of first-order effects plus all two-factor interactions. To this class belong certain $2^{k-p}$ designs with small enough $p$ values. These designs often require rather many scenarios to be simulated. Fortunately, there are also *saturated* designs; i.e., designs with the minimum number of scenarios that still allow unbiased estimators of the regression parameters. Saturated designs are attractive for *expensive* simulations; i.e., simulations that require relatively much computer time per scenario. Saturated resolution V designs were developed by Rechtschaffner (1967).

*Central composite* designs (CCD) are meant for the estimation of second-order polynomials. These designs augment resolution V designs with the base scenario and *2k* scenarios that change factors one at a time; this changing increases and decreases each factor in turn. Saturated variants (smaller than CCD) are discussed in Kleijnen (1987, pp. 314-316).

The main conclusion is that *incomplete designs* for *low-order polynomial regression* are plentiful in both the classic DOE literature and the simulation literature. (The designs in the remainder of this chapter are more challenging.)

*4.4 Designs for simulations with too many factors*

Most practical, non-academic simulation models have many factors; for example, Kleijnen et al. (2003b) experiment with a supply-chain simulation model with nearly 100 factors. Even a Plackett-Burman design would then take 102 scenarios. Because

each scenario needs to be replicated several times, the total computer time may then be prohibitive. For that reason, many analysts keep a lot of factors fixed (at their base values), and experiment with only a few remaining factors. An example is a military (agent-based) simulation that was run millions of times for just a few scenarios—changing only a few factors; see Horne and Leonardi (2001).

However, statisticians have developed designs that require fewer than $k$ scenarios—called *supersaturated designs*; see Yamada and Lin (2002). Some designs *aggregate* the $k$ individual factors into groups of factors. It may then happen that the effects of individual factors cancel out, so the analysts would erroneously conclude that all factors within that group are unimportant. The solution is to define the -1 and +1 levels of the individual factors such that all first-order effects $\beta_j$ ($j = 1, \ldots, k$) are *non-negative*. My experience is that in practice the users do know the direction of the first-order effects of individual factors.

There are several types of group screening designs; for a recent survey including references, I refer to Kleijnen et al. (2003b). Here I focus on the most efficient type, namely *Sequential Bifurcation* designs.

This design type is so efficient because it proceeds *sequentially*. It starts with only two scenarios, namely, one scenario with all individual factors at –1, and a second scenario with all factors at +1. Comparing the outputs of these two extreme scenarios requires only two replications because the aggregated effect of the group factor is huge compared with the intrinsic noise (caused by the pseudorandom numbers). The next step splits— *bifurcates*—the factors into two groups. There are several heuristic rules to decide on how to assign factors to groups (again see Kleijnen et al. 2003b). Comparing the outputs of the third scenario with the outputs of the preceding scenarios enables the estimation of the aggregated effect of the individual factors within a group. Groups—and all its individual factors—are eliminated from further experimentation as soon as the group effect is statistically unimportant. Obviously, the groups get smaller as the analysts proceed sequentially. The analysts stop, once the first-order effects $\beta_j$ of all the important individual factors are estimated. In their supply-chain simulation, Kleijnen et al. (2003b) classify only 11 of the 92 factors as important. (Next, this shortlist of important factors is further investigated to find a robust solution.)

## 5. Kriging

Let's return to the example in Figure 1. If the analysts are interested in the input/output behavior within 'local area 1', then a first-order polynomial may be adequate. Maybe, a second-order polynomial is required to get a valid approximation in 'local area 2', which is larger and shows non-linear behavior of the input/output function. However, Kleijnen and Van Beers (2003a) present an example illustrating that the second-order polynomial gives very poor predictions—compared with Kriging.

Kriging has been often applied in deterministic simulation models. Such simulations are used for the development of airplanes, automobiles, computer chips, computer monitors, etc.; see Sacks et al. (1989)'s pioneering article, and—for an update—see Simpson et al. (2001). For Monte Carlo experiments, I do not know any applications yet. First, I explain the basics of Kriging; then DOE aspects.

### *5.1 Kriging basics*

Kriging is an *interpolation* method that predicts unknown values of a random process; see the classic textbook on Kriging in spatial statistics, Cressie (1993). More precisely, a Kriging prediction is a weighted linear combination of all output values already observed. These weights depend on the distances between the input for which the output is to be predicted and the inputs already simulated. Kriging assumes that *the closer the inputs are, the more positively correlated the outputs are*. This assumption is modeled through the correlogram or the related variogram, discussed below.

Note that in deterministic simulation, Kriging has an important advantage over regression analysis: Kriging is an *exact* interpolator; that is, predicted values at observed input values are exactly equal to the observed (simulated) output values. In random simulation, however, the observed output values are only estimates of the true values, so exact interpolation loses its intuitive appeal. Therefore regression uses OLS, which minimizes the residuals—squared and summed over all observations.

The simplest type of Kriging—to which I restrict myself in this chapter—assumes the following *metamodel* (also see (4) with $\mu = \beta_0$ and $\beta_1 = \beta_2 = \beta_3 = 0$ ):

$$y = \mu + e \text{ with} \tag{13a}$$

$$E(e) = 0, \text{var}(e) = \sigma^2(\boldsymbol{x}) \tag{13b}$$

where $\mu$ is the mean of the stochastic process $y(.)$, and $e$ is the additive noise, which is assumed to have zero mean and non-constant finite variance $\sigma^2(\boldsymbol{x})$ (furthermore, many authors assume normality). Kriging further assumes a *stationary covariance process*; i.e., the expected values $\mu(\boldsymbol{x})$ in (13a) are constant, and the covariances of $y(\boldsymbol{x}+\boldsymbol{h})$ and $y(\boldsymbol{x})$ depend only on the distance (or 'lag') between their inputs, namely $|\boldsymbol{h}| = |(\boldsymbol{x}+\boldsymbol{h}) - (\boldsymbol{x})|$. (In deterministic simulation, the analysts assume that the *deterministic* I/O behavior can be adequately approximated by the *random* model given in (13).)

The Kriging *predictor* for the unobserved input $\boldsymbol{x}_0$ —denoted by $\hat{y}(\boldsymbol{x}_0)$—is a weighted linear combination of all the *n* output data already observed— $y(\boldsymbol{x}_i)$ :

$$\hat{y}(\boldsymbol{x}_0) = \sum_{i=1}^{n} \lambda_i \cdot y(\boldsymbol{x}_i) = \boldsymbol{\lambda}' \cdot \boldsymbol{y} \text{ with} \tag{14a}$$

$$\sum_{\boldsymbol{i}=1}^{\boldsymbol{n}} \lambda_{\boldsymbol{i}} = 1 \tag{14b}$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)^T$ and $\boldsymbol{y} = (y_1, \ldots, y_n)^T$.

To quantify the weights $\boldsymbol{\lambda}$ in (14), Kriging derives the *best linear unbiased estimator* (BLUE), which minimizes the Mean Squared Error (MSE) of the predictor:

$$\text{MSE}\left(\hat{y}(\boldsymbol{x}_0)\right) = E\left(\left(y(\boldsymbol{x}_0) - \hat{y}(\boldsymbol{x}_0)\right)^2\right)$$

with respect to $\boldsymbol{\lambda}$. Obviously, these weights depend on the covariances mentioned below (13). Cressie (1993) characterizes these covariances through the *variogram*, defined as $2\gamma(\boldsymbol{h}) = var(y(\boldsymbol{x}+\boldsymbol{h}) - y(\boldsymbol{x}))$. (I follow Cressie (1993), who uses variograms to express covariances, whereas Sacks et al. (1989) use correlation functions.) It can be proven that the *optimal* weights in (14) are

$$\boldsymbol{\lambda}^T = \left( \boldsymbol{\gamma} + \boldsymbol{1} \frac{\boldsymbol{1} - \boldsymbol{1}^T \boldsymbol{\Gamma}^{-1} \boldsymbol{\gamma}}{\boldsymbol{1}^T \boldsymbol{\Gamma}^{-1} \boldsymbol{1}} \right)^T \boldsymbol{\Gamma}^{-1} \tag{15}$$

with the following symbols:

$\boldsymbol{\gamma}$ : vector of the $n$ (co)variances between the output at the new input $\mathbf{x}_0$ and the outputs at the $n$ old inputs, so $\boldsymbol{\gamma} = (\gamma(\boldsymbol{x}_0 - \boldsymbol{x}_1), \ldots, \gamma(\boldsymbol{x}_0 - \boldsymbol{x}_n))^T$

$\boldsymbol{\Gamma}$ : $n \times n$ matrix of the covariances between the outputs at the $n$ old inputs—with element $(i, j)$ equal to $\gamma(\mathbf{x}_i - \mathbf{x}_j)$

$\boldsymbol{1}$ : vector of $n$ ones.

I point out that the optimal weights defined by (15) vary with the input value for which output is to be predicted (see $\boldsymbol{\gamma}$ ), whereas linear regression uses the same estimated parameters $\hat{\boldsymbol{\beta}}$ for all inputs to be predicted.

*5.2 Designs for Kriging*

The most popular design type for Kriging is *Latin hypercube sampling* (LHS). This design type was invented by McKay, Beckman, and Conover (1979) for deterministic simulation models. Those authors did not analyze the input/output data by Kriging (but they did assume input/output functions more complicated than the low-order polynomials in classic DOE). Nevertheless, LHS is much applied in Kriging nowadays, because LHS is a simple technique (it is part of spreadsheet add-ons such as @Risk).

LHS offers *flexible* design sizes $n$ (number of scenarios simulated) for any number of simulation inputs, $k$. A simplistic example is shown for $k = 2$ and $n = 4$ in Table 5 and Figure 4, which are constructed as follows.

1. The table illustrates that LHS divides each input range into $n$ intervals of equal length, numbered from 1 to $n$ (in the example, we have $n = 4$; see the numbers in the last two columns); i.e., the number of values per input can be much larger than in the designs discussed in Section 4.

2. Next, LHS places these integers 1,…, *n* such that each integer appears exactly once in each row and each column of the design. (This explains the term 'Latin hypercube': it resembles Latin squares in classic DOE.)
3. Within each cell of the design in the table, the exact input value may be sampled uniformly; see Figure 4. (Alternatively, these values may be placed systematically in the middle of each cell. In risk analysis, this uniform sampling may be replaced by sampling from some other distribution for the input values.)

**INSERT Figure 4**
**INSERT Table 5**

Because LHS implies randomness, the resulting design may happen to include *outlier* scenarios (to be simulated). For example, it might happen—with small probability—that in Figure 4 all scenarios lie on the main diagonal, so the values of the two inputs have a correlation coefficient of -1. Therefore LHS may be adjusted to give (nearly) orthogonal designs; see Ye (1998).

Let's compare classic designs and LHS geometrically. Figure 3 illustrates that many classic designs consists of corners of *k*-dimensional cubes. These designs imply simulation of *extreme scenarios*. LHS, however, has better *space filling* properties.

This property has inspired many statisticians to develop other space filling designs. One type maximizes the minimum Euclidean distance between any two points in the *k*-dimensional experimental area. Related designs minimize the maximum distance. See Koehler and Owen (1996), Santner et al. (2003), and also Kleijnen et al. (2003a).

## 6. Conclusions

Because simulation—treated as a black box—implies *experimentation* with a model, design of experiment is essential. In this chapter, I discussed both *classic* designs for low-order polynomial regression models and *modern* designs (including Latin hypercube sampling) for other metamodels such as Kriging models. The simpler the metamodel is, the fewer scenarios need to be simulated. (Cross validation of the metamodel selected, is discussed in Chapter III.1 by Wang.)

I did not discuss so-called *optimal designs* because these designs use statistical assumptions (such as white noise) that I find too unrealistic. A recent discussion of optimal designs including references is Spall (2003).

Neither did I discuss the designs in *Taguchi* (1987), as I think that the classic and modern designs (which I did discuss) are superior. Nevertheless, I think that Taguchi's concepts—as opposed to his statistical techniques—are important. In practice, the 'optimal' solution may break down because the environment turns out to differ from the environment that the analysts assumed when deriving the optimum. Therefore they should look for a 'robust' solution. For further discussion I refer to Kleijnen et al. (2003a).

Because of space limitations, I did not discuss *sequential* designs, except for sequential bifurcation and two-stage resolution IV designs. Nevertheless, the sequential nature of simulation experiments (caused by the computer architecture) makes sequential designs very attractive. This is an area of active research nowadays; see Jin et al. (2002), Kleijnen et al. (2003a), and Kleijnen and Van Beers (2003b).

I mentioned several more research issues; for example, importance sampling. Another interesting question is: how much computer time should analysts spend on *replication*; how much on exploring *new* scenarios?

Another challenge is to develop designs that explicitly account for *multiple outputs*. This may be a challenge indeed in sequential bifurcation (depending on the output selected to guide the search, different paths lead to the individual factors identified as being important). In practice, multiple outputs are the rule in simulation; see Kleijnen et al. (2003a).

The application of *Kriging* to *random* simulation models (such models are a focus of this handbook, including this chapter) seems a challenge. Moreover, corresponding software needs to be developed. Current software focuses on deterministic simulation; see Lophaven et al. (2002).

Comparison of various metamodel types and their designs remains a major problem. For example, Meckesheimer et al. (2001) compare radial basis, neural net, and polynomial metamodels. Clarke et al. (2003) compare low-order polynomials, radial basis functions, Kriging, splines, and support vector regression. Alam et al. (2003) found that LHS gives the best neural-net metamodels. Comparison of screening designs has hardly begun; see Kleijnen et al. (2003 a, b).

**References**

Alam, F.M., K.R. McNaught, T.J. Ringrose. 2003. A comparison of experimental designs in the development of a neural network simulation metamodel. *Simulation Modelling: Practice and Theory*, accepted conditionally.

Angün, E., D. den Hertog, G. Gürkan, J.P.C. Kleijnen. 2002. Response surface methodology revisited. In: *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yücesan, C.H. Chen, J.L. Snowdon, J.M. Charnes, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, 377-383.

Antioniadis, A., D.T. Pham. 1998. Wavelet regression for random or irregular design. *Computational Statistics and data Analysis* **28** 353-369.

Bettonvil, B., J.P.C. Kleijnen. 1990. Measurement scales and resolution IV designs. *American Journal of Mathematical and Management Sciences* 10 (3-4): 309-322.

Clarke, S.M., J.H Griebsch, T.W., Simpson. 2003. Analysis of support vector regression for approximation of complex engineering analyses. *Proceedings of DETC '03, ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago*.

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

Donohue, J. M., E. C. Houck, R.H. Myers. 1993. Simulation designs and correlation induction for reducing second-order bias in first-order response surfaces *Operations Research* 41 (5): 880-902.

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

Glasserman, P., P. Heidelberger, and P. Shahabuddin (2000), Variance reduction techniques for estimating value-at-risk. *Management Science*, 46, no. 10, pp. 1349-1364

Horne, G., M. Leonardi, eds. 2001. *Maneuver warfare science 2001*. Quantico, Virginia: Defense Automatic Printing Service

Jin, R, W. Chen, and A. Sudjianto (2002), On sequential sampling for global metamodeling in engineering design. *Proceedings of DETC '02, ASME 2002 Design Engineering Technical Conferences and Computers and Information*

*in Engineering Conference*, DETC2002/DAC-34092, September 29-October 2, 2002, Montreal, Canada

Kleijnen, J.P.C. 1998. Experimental design for sensitivity analysis, optimization, and validation of simulation models. In: *Handbook of Simulation*, ed. J. Banks, 173-223. New York, Wiley

Kleijnen, J.P.C. 1992. Regression metamodels for simulation with common random numbers: comparison of validation tests and confidence intervals. *Management Science* 38 (8): 1164-1185.

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

Kleijnen, J.P.C. 1975. *Statistical techniques in simulation*, *volume II*. New York: Marcel Dekker. (Russian translation, Publishing House "Statistics", Moscow, 1978.)

Kleijnen, J.P.C. S.M. Sanchez, T.W. Lucas, T.M. Cioppa. 2003a. A user's guide to the brave new world of designing simulation experiments. *INFORMS Journal on Computing* (accepted conditionally)

Kleijnen, J.P.C., B. Bettonvil, F. Person. 2003b. Finding the important factors in large discrete-event simulation: sequential bifurcation and its applications. In: *Screening*, ed. A.M. Dean, S.M. Lewis, New York: Springer-Verlag (forthcoming; preprint: http://center.kub.nl/staff/kleijnen/papers.html).

Kleijnen, J.P.C. and R.Y. Rubinstein (2001), Monte Carlo sampling and variance reduction techniques. *Encyclopedia of Operations Research and Management Science*, Second edition, edited by S. Gass and C. Harris, Kluwer Academic Publishers, Boston, 2001, pp. 524-526

Kleijnen, J.P.C., 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 2003a. Robustness of Kriging when interpolating in random simulation with heterogeneous variances: some experiments. *European Journal of Operational Research* (accepted conditionally).

Kleijnen, J.P.C., W.C.M. Van Beers. 2003b.Application-driven sequential designs for simulation experiments: Kriging metamodeling. Working Paper, CentER, Tilburg University, Tilburg, The Netherlands (submitted for publication; preprint: http://center.kub.nl/staff/kleijnen/papers.html)

Koehler, J.R., A.B. Owen. 1996. Computer experiments. In: *Handbook of Statistics, Volume 13*, eds. S. Ghosh, C.R. Rao, 261-308. Amsterdam: Elsevier.

Law, A.M., W.D. Kelton. 2000. *Simulation modeling and analysis*. 3rd ed. McGraw-Hill, New York

Lophaven, S.N., H.B. Nielsen, and J. Sondergaard (2002), *DACE: a Matlab Kriging toolbox, version 2.0*. IMM Technical University of Denmark, Lyngby

McKay, M.D., R.J. Beckman, W.J. Conover. 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21 (2): 239-245 (reprinted in 2000: *Technometrics* 42 (1): 55-61).

Meckesheimer, M., R.R. Barton, F. Limayem, B. Yannou. 2001. Metamodeling of combined discrete/continuous responses. *AIAA Journal* 39 1950-1959.

Rechtschaffner, R.L. 1967. Saturated fractions of 2n and 3n factorial designs. *Technometrics* **9** 569-575.

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

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

Simpson, T.W., T.M. Mauery, J.J. Korte, F. Mistree. 2001. Kriging metamodels for global approximation in simulation-based multidisciplinary design optimization. *AIAA Journal* 39 (12) 2233-2241.

Spall, J.C. 2003. *Introduction to stochastic search and optimization; estimation, simulation, and control*. Hoboken, New Jersey, Wiley.

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

Yamada, S., Lin, D. K. J. (2002). Construction of mixed-level supersaturated design, *Metrika* (56), 205-214. (available online: http://springerlink.metapress.com/app/home/)

Ye, K.Q. (1998), Orthogonal column Latin hypercubes and their application in computer experiments. *Journal Association Statistical Analysis, Theory and Methods*, (93) 1430-1439.

Zeigler B.P., K. Praehofer, T.G. Kim. 2000. *Theory of modeling and simulation. 2nd ed.* New York: Academic Press.

Table 1. A one-factor-at-a-time design for two factors, and possible regression variables

| scenario | $x_0$ | $x_1$ | $x_2$ | $x_1x_2$ |
|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 |
| 2 | 1 | 1 | 0 | 0 |
| 3 | 1 | 0 | 1 | 0 |

Table 2. The $2^3$ design and possible regression variables

| Scenario | 0 | 1 | 2 | 3 | 1.2 | 1.3 | 2.3 | 1.2.3 |
|---|---|---|---|---|---|---|---|---|
| 1 | + | - | - | - | + | + | + | - |
| 2 | + | + | - | - | - | - | + | + |
| 3 | + | - | + | - | - | + | - | + |
| 4 | + | + | + | - | + | - | - | - |
| 5 | + | - | - | + | + | - | - | + |
| 6 | + | + | - | + | - | + | - | - |
| 7 | + | - | + | + | - | - | + | - |
| 8 | + | + | + | + | + | + | + | + |

Table 3. A $2^{7-4}$ design

| scenario | 1 | 2 | 3 | 4 = 1.2 | 5= 1.3 | 6 = 2.3 | 7 = 1.2.3 |
|---|---|---|---|---|---|---|---|
| 1 | - | - | - | + | + | + | - |
| 2 | + | - | - | - | - | + | + |
| 3 | - | + | - | - | + | - | + |
| 4 | + | + | - | + | - | - | - |
| 5 | - | - | + | + | - | - | + |
| 6 | + | - | + | - | + | - | - |
| 7 | - | + | + | - | - | + | - |
| 8 | + | + | + | + | + | + | + |

Table 4. The Placket-Burman design for 11 factors

| scenario | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | + | - | + | - | - | - | + | + | + | - | + |
| 2 | + | + | - | + | - | - | - | + | + | + | - |
| 3 | - | + | + | - | + | - | - | - | + | + | + |
| 4 | + | + | + | - | + | + | - | - | - | + | + |
| 5 | + | - | + | + | - | - | - | - | - | - | + |
| 6 | + | + | - | + | + | + | + | + | - | - | - |
| 7 | - | + | + | + | - | + | + | - | + | - | - |
| 8 | - | - | + | + | + | - | + | + | - | + | - |
| 9 | - | - | - | + | + | + | - | + | + | - | + |
| 10 | + | - | - | - | + | + | + | - | + | + | - |
| 11 | - | + | - | - | - | + | + | + | - | + | + |
| 12 | - | - | - | - | - | - | - | - | - | - | - |

Table 5. A LHS design for two factors and four scenarios

| Scenario | Interval factor 1 | Interval factor 2 |
|---|---|---|
| 1 | 2 | 1 |
| 2 | 1 | 4 |
| 3 | 4 | 3 |
| 4 | 3 | 2 |

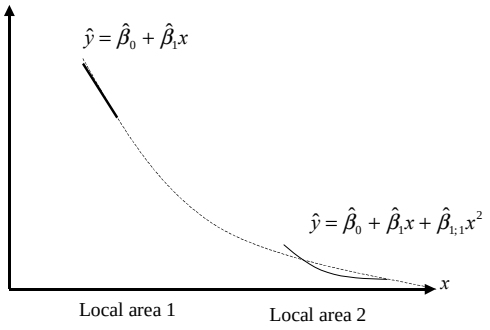


Fig. 1: Two simple polynomial regression models with predictor $\hat{y}$ for the output of a simulation with a single factor $x$

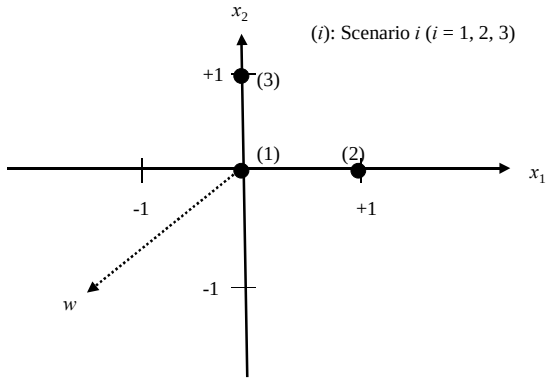


Fig. 2: One-factor-at-a-time design for two factors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, with output $w$

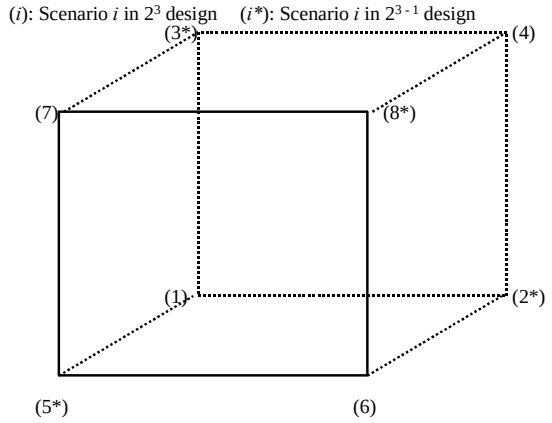


Fig. 3: The $2^3$ design

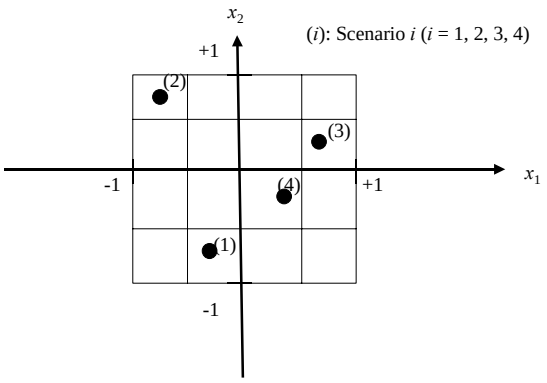


Fig. 4. A LHS design for two factors and four scenarios