Statistical Testing of Optimality Conditions in Multiresponse Simulation-Based Optimization (Replaced by Discussion Paper 2007-45)

Bettonvil, B.W.M.; Del Castillo, E.; Kleijnen, Jack P.C.

Publication date:
2005

Citation for published version (APA):
Bettonvil, B. W. M., Del Castillo, E., & Kleijnen, J. P. C. (2005). *Statistical Testing of Optimality Conditions in Multiresponse Simulation-Based Optimization (Replaced by Discussion Paper 2007-45)*. (CentER Discussion Paper; Vol. 2005-81). Operations research.

No. 2005–81

STATISTICAL TESTING OF OPTIMALITY CONDITIONS IN MULTIRESPONSE SIMULATION-BASED OPTIMIZATION

By Bert Bettonvil, Enrique del Castillo, Jack P.C. Kleijnen

June 2005

ISSN 0924-7815

TILBURG UNIVERSITY

Statistical testing of optimality conditions in multiresponse simulation-based optimization

Bert Bettonvil [1], Enrique del Castillo [2], and Jack P.C. Kleijnen [1, 3]

[1] Department of Information Systems & Management/Center for Economic Research
Tilburg University, Postbox 90153, 5000 LE Tilburg, Netherlands

[2] Department of Industrial & Manufacturing Engineering & Department of Statistics, The Pennsylvania State University, 310 Leonhard building, University Park PA 16802, USA

[3] Operations Research & Logistics Group/Mansholt Graduate School of Social Sciences
Wageningen University and Research Centre, Hollandseweg 1, 6706 KN Wageningen, Netherlands

Abstract: This paper derives a novel procedure for testing the Karush-Kuhn-Tucker (KKT) first-order optimality conditions in models with multiple random responses. Such models arise in simulation-based optimization with multivariate outputs. This paper focuses on 'expensive' simulations, which have small sample sizes. The paper estimates the gradients (in the KKT conditions) through low-order polynomials, fitted locally. These polynomials are estimated using Ordinary Least Squares (OLS), which also enables estimation of the variability of the estimated gradients. Using these OLS results, the paper applies the bootstrap (resampling) method to test the KKT conditions. Furthermore, it applies the classic Student *t* test to check whether the simulation outputs are feasible, and whether any constraints are binding. The paper applies the new procedure to both a synthetic example and an inventory simulation; the empirical results are encouraging.

Keyword: Stopping rule; metaheuristics; RSM, design of experiments

JEL: C0, C1, C9C 15, C44, C61, C9

1. Introduction

In this paper we present a novel multi-stage procedure to test whether a given input combination (also called factor combination, scenario, or iterate) for a random simulation model with multiple responses (also called multivariate outputs) is optimal (we give several synonyms, because simulation-based optimization is studied in many disciplines, each with its own terminology). In this way, our test provides a stopping criterion for iterative, heuristic simulation-based optimization. We reason as follows.

(i) The *Karush-Kuhn-Tucker* (KKT) first-order optimality conditions are well known in deterministic nonlinear mathematical programming (see, for example, Gill, Murray, and Wright 2000, p. 81). We shall formalize these conditions in equation (2).

(ii) These KKT conditions may be checked in *random* simulation, by means of *asymptotic* tests (based on the delta method). By definition, these tests assume large numbers of 'replicates'; replicates mean that a particular scenario is simulated several times, using non-overlapping streams of pseudo-random numbers (PRN). Details are given by Angűn and Kleijnen (2004), Shapiro (2000), and Shapiro and Homem-de-Mello (1998).

(iii) We, however, assume that the simulation model at hand is so *expensive* that we generate a single replicate for each simulated input combination—except for the (single) center point of the design that specifies the input combinations to be simulated, which is replicated a few times (each simulation output consists of multiple responses).

There are many methods for optimizing simulated systems (see, for example, the survey paper Fu 2002 or the monograph Spall 2003). Many methods ignore the fact that in practice simulation models generate *multiple* responses per scenario. For example, an academic (s, S) inventory simulation—with reorder level s and order-up-to quantity S (so there are two inputs)—defines *the* output as the expected (or mean) sum of the inventory-carrying, ordering, and out-of-stock costs, whereas a practical simulation typically has two responses, namely the sum of the average inventory-carrying and ordering costs —which is to be minimized—and the service probability (also called the fill rate)—which must satisfy a prespecified lower bound (say, 95%). In this paper, we select one of the multiple responses and minimize that response, while we satisfy constraints on the remaining random (noisy) responses. Figure 1 illustrates this problem (details will follow in Section 2). This figure demonstrates that the KKT conditions require the estimation of the local *gradient* of response h (say) $\boldsymbol{\beta}_{-0;\,h}$ (there are z responses so h = 0, 1, …, z – 1 with 0 denoting the goal response, which is to be minimized; the subscript –0 denotes the elimination of the intercept $\beta_{0;\,h}$, which we estimate automatically when applying OLS analysis; we suppress the symbol $\boldsymbol{d}$ in $\boldsymbol{\beta}_{-0;\,h}(\boldsymbol{d})$ where $\boldsymbol{d}$ implies a specific local area; see Section 2).

Insert Figure 1: An example of a constrained nonlinear random optimization problem

Some of these optimization methods treat the simulation model as a *black box*; i.e., they observe Input/Output (I/O data only (see again Fu 2002 and Spall 2003). Examples are the many meta-heuristics (ant colony optimization, genetic and evolutionary algorithms, scatter search, simulated annealing, tabu search), including response surface methodology (RSM). Other methods treat the simulation as a *white box*, so they can estimate the gradients from a single simulation run. Best known are perturbation analysis and the score function (or likelihood ratio) method. Our procedure can be combined with any method that estimates the gradient—either from a single run or from several runs—provided the method also estimates the density function (distribution) of the gradient estimator, as we shall see below.

Note: To check the KKT conditions, Karaesman and Van Ryzin (2004) present an *un*constrained optimization algorithm that uses the estimated gradient of the goal function, including a score function estimator.

We use this Estimated Density Function (EDF) of the estimated gradients for *bootstrapping*. In general, the bootstrap can estimate the distribution of *any* statistic provided that the likelihood function is continuous; see the seminal book on bootstrapping (outside simulation), Efron and Tibshirani (1993, pp. 54-56, 162-177). But, those authors caution: ‘bootstrapping is not a uniquely defined concept [...] alternative bootstrap methods may coexist’ (see Efron and Tibshirani 1993, pp. 115, 383). Moreover, we wish to test the *hypothesis* that a specific input combination satisfies the KKT conditions—and Shao and Tu (1995, p. 189) warn: ‘bootstrap hypothesis testing ... is not a well-developed topic’.

We apply our procedure to the following two examples:

(i) A synthetic (artificial, numerical, Monte Carlo) example so we—but not our method—know its I/O function explicitly.

(ii) An inventory simulation that has only an estimated implicit I/O function.

Our empirical results are encouraging; i.e., the type I error rates are close to the prespecified (nominal) rates; the type II error rates (complement of the power) decrease as the input combination tested moves farther away from the true optimum.

The remainder of this paper is organized as follows. Section 2 formalizes a constrained nonlinear random optimization problem, and its KKT conditions. Section 3 uses OLS to locally fit either a first-order or a second-order polynomial per response, using either a Resolution-3 (R-3) design augmented with a center point or a Central Composite Design (CCD). Section 4 develops a procedure for testing whether the center of the local area satisfies the KKT conditions. Its subsection 4.1 uses Student’s *t* test to check whether the simulation responses are feasible, and whether any constraints are binding (active). Subsection 4.2 derives a bootstrap procedure to test the remaining KKT conditions. Section 5 studies the performance of the novel procedure by means of the synthetic example. Section 6 illustrates the procedure through its application to an inventory simulation. Section 7 gives conclusions and future research topics.

2. Mathematical programming formulation of simulation optimization with multivariate outputs

Following Angün et al. (2002), we formalize our problem as follows. The simulation model has $k \geq 1$ inputs. Let d_j denote the value of the original (non-standardized) input *j* (j = 1, …, k). Let the *z* responses be denoted by $w_{h'}$, (h'= 0, …, z – 1), where the goal output—to be minimized—corresponds with w_0. This results in the following *constrained nonlinear random optimization problem*:

$$\begin{array}{lll} \text{Minimize} & \mathrm{E}(w_0(\boldsymbol{d},\boldsymbol{r})) & \\ \text{subject to} & \mathrm{E}(w_h(\boldsymbol{d},\boldsymbol{r})) \geq a_h \text{ for } h = 1, ..., z-1 & (1) \end{array}$$

where $\boldsymbol{r}$ denotes the PRN and a_h denotes the right-hand-side value for constraint *h*.

Figure 1 (displayed in Section 1) illustrates (1). This figure has two inputs $\boldsymbol{d} = (d_1, d_2)^T$; see the labels of the two axes. Furthermore, this figure has three outputs $\boldsymbol{w} = (w_0, w_1, w_2)^T$; see the labels of the various contour functions. Actually, the figure shows (only) three 'iso' goal functions, which (by definition) are the set of input combinations with the same goal value—namely, 96, 76, and 66 respectively. The figure shows two constraints, namely $\mathrm{E}(w_1) = 4$ and $\mathrm{E}(w_2) = 9$. The optimal input (to be found by some given simulation optimization procedure) is point A; three suboptimal points are also shown—namely B, C, and D—with binding constraints 1 and 2 respectively. The figure also shows the gradients of the goal function and the constraint that is binding in the specific location; these gradients are (by definition) perpendicular to the local tangent lines, but those lines are shown only for the binding constraint (not for the goal function).

The well-known *KKT conditions* for (deterministic) problem (1) are

$$\boldsymbol{\beta}_{-0;\,0} = \boldsymbol{B}_{-0;\,J}\boldsymbol{\lambda} \qquad (2)$$

where $\boldsymbol{\beta}_{-0;\,0}$ denotes the (deterministic) gradient of the goal function (also see $\boldsymbol{\beta}_{-0;\,h}$ defined in Section 1); $\boldsymbol{B}_J$ is the $k \times J$ matrix with the gradients of the *J binding* constraints, and $\boldsymbol{\lambda}$ denotes the corresponding *non-negative* Lagrange multipliers. For example, Figure 1 shows that point A satisfies (2), as $\boldsymbol{\beta}_{-0;\,0}$ and $\boldsymbol{B}_1$ point in (roughly) the same direction. Points B and C have $\boldsymbol{\beta}_{-0;\,0}$ and $\boldsymbol{B}_1$ point in different but similar directions. Point D has $\boldsymbol{\beta}_{-0;\,0}$ and $\boldsymbol{B}_2$ point in completely different directions. Note that at these four points (A through D) the matrix $\boldsymbol{B}_1$ has only one column; this column consists of the components of the gradient of the constraint that is binding at the specific point.

Note: If the optimum occurs *inside* the feasible area, there are no binding constraints. Then the KKT conditions reduce to the condition that the goal gradient is zero. The latter condition may be tested through a classic *F*-test; see the classic RSM or the linear regression literature (also see again Karaesman and Van Ryzin 2004). We do not consider this case any further.

Unfortunately, in *random* simulation the gradients must be estimated. Moreover, the slacks of the constraints must be estimated, to check which constraints are binding. This estimation turns the KKT conditions (2) into a problem of nonlinear statistics—discussed next.

3. **Estimation of gradients in random simulation**

In this section, we show how we may estimate the gradients in random simulation, using
(i) a proper experimental design to select the input combinations to be simulated;
(ii) OLS to analyze the resulting I/O simulation data.

Many analysts estimate the gradients in black-box (either random or deterministic) simulations by *changing one input at a time*, followed by some type of differencing; see Spall (2003). We, however, propose to estimate the coefficients (parameters) of either a first-order or a second-order polynomial—locally fitted per response (also see the tangent lines in Figure 1 above). The statistical theory on Design Of Experiments (DOE) proves that the best design to estimate a *first-order polynomial* is an *R-3 design*, which requires only $n = k + 1$ input combinations with $k + 1$ rounded upwards to the next multiple of four (remember: k denotes the number of inputs). For example, if $4 \leq k \leq 7$ then $n = 8$, which is the number of combinations in a (fractional two-level factorial) 2^{7-4} design. These 2^{k-p} fractional designs ($-p$ denotes the fraction) are a subclass of the Plackett-Burman designs. For example, if $8 \leq k \leq 11$ then $n = 12$, which is not a 2^{k-p} design. We refer to Kleijnen (1987) and Myers and Montgomery (2002) for details. Figure 1 has only two inputs so $n = 2^2$, which corresponds with a full factorial design.

Joshi, Sherali, and Tew (1998) use *second-order polynomials* to estimate (conjugate) gradients (but they do not test the KKT conditions; classic RSM assumes that the single response reaches its maximum at a ‘hill top’, which is modeled through a second-order polynomial). To estimate a second-order polynomial, different designs are available; again see Kleijnen (1987) and Myers and Montgomery (2002). The most popular design, however, is the CCD, which consists of the following subdesigns:

i. A *Resolution-5* (R-5) design, which—by definition—enables unbiased estimation of the k main effects, the $k(k – 1)/2$ two-factor interactions (or cross-products in the polynomial), and the intercept—provided no other effects are important (however, we assume that purely quadratic effects are important). In Figure 1 and our synthetic example there are only two factors, so the 2^2 design is an R-5 design.

ii. The $2k$ *axial* points, which means that each factor j (j = 1, …, k) is simulated at the value (say) $–c$ and $+c$ while all other (k - 1) factors are at their base value (zero). These points enable estimation of the purely quadratic effects.

iii. The *center* point, which is replicated. This replication is used to test the fit of the estimated polynomial (as we shall see below).

Note that the CCD makes the estimated purely quadratic effects and intercept correlated. To avoid singularity in the OLS estimator defined in (3) below, the design should satisfy the condition $n \geq q$; actually the CCD are not saturated at all: $n >> q$ (also see the artificial and the inventory examples with two inputs, discussed in Sections 5 and 6).

The design determines the input combinations that are actually simulated. After this simulation, the parameters of the polynomials are estimated. To estimate these parameters, classic DOE uses OLS. The OLS assumptions imply that the ‘fitting error’ (also called ‘error’ or ‘disturbance’) is *white noise*; i.e., these errors (say) e are Normally, Identically, and Independently Distributed (NIID) with zero mean ($\mu_e = 0$) and ‘constant’ variance (σ_e^2); i.e., the variance is locally constant (but not globally: for example, the local areas centered around the points A, B, C, and D in Figure 1 may have different variances). The error e represents the joint effects of (i) lack of fit, and (ii) intrinsic variation caused by the use of PRN in random simulation. Furthermore, classic DOE assumes a univariate output; we shall discuss this issue in the paragraph including (3).

We now define some more symbols. Each first-order polynomial has k ‘main effects’ β_j ($j = 1,...,k$)—if there are k inputs—and an ‘intercept’ or ‘grand mean’ β_0, which together define the regression parameter vector $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)^T$. Its OLS estimator is denoted by $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k)^T$, which implies the *estimated local gradient* $\hat{\boldsymbol{\beta}}_{-0} = (\hat{\beta}_1, \ldots, \hat{\beta}_k)^T$. Note that this gradient is *biased* if higher order effects are important and the design does not protect against this bias; for example, by definition, Resolution-4 (R-4) designs protect against bias caused by two-factor interactions, but not against purely quadratic effects (see next paragraph).

A second-order polynomial has the first-order effects $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)^T$ plus the 'purely quadratic effects' (say) $\beta_{j;\,j}$ $(j = 1, ..., k)$ and the two-factor interactions $\beta_{j;\,j'}$ $(j' > j)$. Altogether this polynomial has a regression parameter vector with (say) q parameters, $\boldsymbol{\beta}_{full} = (\beta_0, \beta_1, \ldots, \beta_{q-1})^T$ (the subscript 'full' denotes the full model, which should be distinguished from the 'reduced' model that has no second-order effects). Obviously, the OLS estimator of these parameters is denoted by $\hat{\boldsymbol{\beta}}_{full} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_{q-1})^T$. This implies the *estimated local gradient* with components $\partial y / \partial x_j =$ $\hat{\beta}_j + \hat{\beta}_{j;\,j'} x_{j'} + 2\hat{\beta}_{j;\,j} x_j$. Because we estimate the gradient at the center point (where x_j = 0), the estimated gradient reduces to $\hat{\boldsymbol{\beta}}_{-0} = (\hat{\beta}_1, \ldots, \hat{\beta}_k)^T$. This is the same expression as we derived for the first-order polynomial, but obviously the estimates are different; i.e., the estimated gradient is biased if second-order effects are important and yet a first-order polynomial is used.

Whereas classic DOE assumes a single response per input combination, we assume *multiple* responses $w_{h'}$ (h' = 0, 1, … z - 1); see (1). Like Angün et al. (2003, 2002), we first fit a local *first-order polynomial* for each of these z responses; unlike Angün et al. we also consider second-order polynomials (if the first-order polynomial gives significant lack-of-fit; see equation 6 below). Like classic DOE and Angün et al., we further assume that these z responses together form a *multivariate Gaussian* variate. This assumption is realistic if the simulation responses are averages so some limit theorem applies; for example, in our inventory simulation the two responses are costs and service percentages averaged over very many periods (see Section 6). A multivariate Gaussian variate is characterized by its vector of z means $\mathrm{E}(w_{h'}(\boldsymbol{d}, \boldsymbol{r}))(h' = 0, ..., z-1)$ and its $z \times z$ covariance matrix $\hat{\boldsymbol{cov}}(w_{h'}, w_{h''})$ (h', h'' = 0, 1, … z - 1). The z responses for a specific scenario are correlated (so $\hat{\boldsymbol{cov}}(w_{h'}, w_{h''})$ is not a diagonal matrix), since they map the same PRN using different transformation functions; for example, our inventory simulation will record the average ordering costs and service percentage per scenario. Furthermore, the z responses have different variances; in the inventory example, the ordering costs and the service percentage have different dimensions!

Because the simulation output is multivariate, the *Best Linear Unbiased Estimator* (BLUE) of the regression parameters $\boldsymbol{\beta}$ seems to require Generalized Least Squares (GLS) instead of OLS. However, since all z responses use the same design, the GLS estimator reduces to the OLS estimator (see Rao 1967 and for a more recent publication see Ruud 2000, p. 703):

$$\hat{\boldsymbol{\beta}}_{h'} = (\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{w}_{h'} \text{ with } h' = 0, ..., \quad z-1 \tag{3}$$

where $\boldsymbol{X}$ denotes the $n \times q$ matrix of explanatory (regression) variables. This $\boldsymbol{X}$ is completely determined by $\tilde{\boldsymbol{D}}$, the *standardized* design matrix for the k inputs—with elements $\tilde{d}_{i;\,j}$ that are linear transformations of the original $d_{i;\,j}$ in (1) such that the $\tilde{d}_{i;\,j}$ of the R-3 design lie between -1 and +1 (i = 1, …, n and j = 1, …, k). This standardization simplifies our computations; for example, the matrix in (3) to be inverted becomes a diagonal matrix in case of an R-3 design analyzed through a first-order polynomial. If we use a second-order polynomial, then we use a CCD with axial points determined by the constant c. The selection of a particular value for this c is discussed in the classic DOE literature, given specific assumptions. For our examples with only two factors we use $c = \sqrt{2}$; see Myers and Montgomery (1995, p. 298).

Note: Angün et al. (2003, 2002) use an R-3 design, assuming that a first-order polynomial is adequate. They simulate one of the input combinations of the R-3 design two times, and all remaining combinations only once; their selection of the combination to be replicated is arbitrary.

Besides obtaining the point estimates of the gradients through (3), we wish to obtain the *estimated covariance matrix* of these estimated gradients. We again consider first-order and second-order polynomials respectively.

If we assume first-order polynomials, then a non-replicated R-3 design (which minimizes computer time in expensive simulation) enables unbiased estimators of the covariance matrixes of the z individual gradients—provided this design is not *saturated* (a saturated design implies that all residuals are zero). The design is saturated if $k + 1$ (k denotes the number of inputs) equals a multiple of four (so $k = 3, 7, 11, 15, \ldots.$). Even if the design is not saturated, we propose to simulate—besides the R-3 design—the *center point* of the local experimental, to test whether a constraint is *binding* in a local area; the center point is more representative of the local behavior than any of the corner points that are part of the R-3 design (also see Section 4.1).

The R-3 design enables unbiased estimators of gradients and their covariance matrices, provided a first-order polynomial is *adequate*. In practice it is not obvious how small the local area should be selected, to make this polynomial adequate. Therefore this adequacy is tested through a lack-of-fit F statistic. This statistic requires that one or more points be replicated, so that *pure error* can be estimated. Because we focus on expensive simulations, we propose to minimize the number of replicates. Therefore we replicate only the center point (as is traditional in RSM). The classic univariate statistic requires that this point be observed at least twice: $m \geq 2$ where m denotes the number of replicates at the center of the local area. The multivariate statistic defined by Roy, Gnanadesikan, and Srivastava (1971, p. 35)—also see Angűn and Kleijnen (2004), Dykstra (1959) and Khuri (1996, p. 385)— requires $m \geq z + 1$ to obtain a non-singular estimated covariance matrix based on replications (so if $z = 1$, then the multivariate statistic requires the same m as the univariate). For example, in the synthetic example we have $z = 3$ responses, so we take $m = 4$ observations at the local center point; in the inventory illustration we have $z = 2$ so $m = 3$ (Kleijnen (1993) presents a case study with 2 response types and as many as 14 inputs, so m would have to be 3 at least).

The lack-of-fit test compares two variance estimators:

(i) one estimator based on the *residuals* $(\boldsymbol{w}_{h'} - \hat{\boldsymbol{y}}_{h'})$ with $\boldsymbol{w}_{h'}$ defined in (1) and $\hat{\boldsymbol{y}}_{h'} = \boldsymbol{X}\hat{\boldsymbol{\beta}}_{h'}$, and

(ii) one estimator based on replication.

Obviously, only the first estimator depends on the polynomial (regression metamodel) that is selected to approximate the true I/O function implied by the underlying simulation model.

The *Mean Squared Residual* (MSR) estimator of the covariance matrix $\boldsymbol{cov}(w_{h'}, w_{h''})$ is

$$\hat{\boldsymbol{cov}}_{residuals}(\boldsymbol{w}_{h'}, w_{h''}) = \left(\frac{(\boldsymbol{w}_{h'} - \hat{\boldsymbol{y}}_{h'})^T (\boldsymbol{w}_{h''} - \hat{\boldsymbol{y}}_{h''})}{N - q} \right) (h', h'' = 0, \ldots, \quad z - 1) \quad (4)$$

where $N = \sum_{i=1}^{n} m_i$ denotes the total number of simulation runs with m_i denoting the number of replicates for scenario *i,* and $\hat{\boldsymbol{y}}_{h'} = \boldsymbol{X}\hat{\boldsymbol{\beta}}_{h'}$ is the vector with the *N* regression predictors of simulation responses $\boldsymbol{w}_{h'}$ for the *N* scenarios *i* (obviously, replicated scenarios have the same predictor because they have the same input). All *N* scenarios use different, non-overlapping PRN to make the MSR unbiased; i.e., we do not use Common Random Numbers (CRN). We select all m_i equal to one—except for the center point, which has $m \geq z + 1$ replicates.

We now consider two cases of (4):

(i) In case of a specific response $h' = h''$, (4) gives (say) $\hat{\sigma}_{h';h'} = \hat{\sigma}^2_{h'}$, which is the classic MSR for that response; i.e., it estimates the variance of simulation response $w_{h'}$. (Remember: this variance is assumed to be constant within the local area.)

(ii) In case of two different responses $h' \neq h''$, (4) estimates the covariance between the responses $w_{h'}$ and $w_{h''}$. (This covariance is again assumed to be constant within the local area.) This covariance is used by the multivariate *F*-statistic defined by Roy et al. (1971), but not by the classic univariate *F*-statistic. The latter statistic is simpler, requires fewer replications, and may be combined with Bonferroni's inequality to hedge against non-zero covariances—albeit that the use of this inequality makes the test 'conservative'; i.e., the test has a smaller type-I error rate than prespecified. (We shall also use these covariances to estimate the covariances between the estimated gradients of different responses; see (7).)

The estimator based on *replication* of the center point is

$$\hat{\boldsymbol{cov}}_{replicates}(w_{h'}, w_{h''}) = \left(\frac{\sum_{r=1}^{m}(w_{h';r} - \overline{w}_{h'})(w_{h'';r} - \overline{w}_{h''})}{m-1} \right) \tag{5}$$

where $w_{h';r}$ denotes response h' of replication *r*—with *r* = 1, …, ,*m*—of the (local) center point, and $\overline{w}_{h'}$ denotes the average of these *m* responses ; we suppress the subscript for the scenario.

The classic (univariate) *lack-of-fit test* is

$$F_{n-q;\,N-n}(h') = \frac{\sum_{i=1}^{n} m_i(\overline{w}_{i;h'} - \hat{y}_{i;h'})^2 /(n-q)}{\sum_{i=1}^{n}\sum_{r=1}^{m_i} (w_{i;h';r} - \overline{w}_{i;h'})^2 /(N-n)} \tag{6}$$

where $w_{i;h';r}$ denotes response h' in replication *r* of scenario *i*; see Myers and Montgomery (1995, p. 52).

If this statistic is significant (using Bonferroni's inequality), then the MSR estimator in (4) overestimates the true variance. For example, if the true response variance is zero (as in deterministic simulation), then (4) still gives a positive variance estimate in case the polynomial does not fit perfectly.

If we find significant lack of fit, we have two options:

(i) Decrease the local area; for example, halve each factor's range.

(ii) Increase the order of the polynomial; for example, switch from a first-order to a second-order polynomial.

If we do not find significant lack of fit, then we will still base our bootstrap on the replicates because the MSR estimator may be inflated by undetected bias (the lack-of-fit test has small power if the number of replicates is small).

Note: We might use CRN, to reduce the noise of the estimated gradients. Unfortunately, we must then estimate the covariances between $\boldsymbol{w}_{h';i}$ and $\boldsymbol{w}_{h'';i'}$ (with $\boldsymbol{w}_{h'';i'}$ denoting response h'' at scenario i', and i' = 1, …, *n*). This leads from the $z \times z$

covariance matrix in (5) to a $zn \times zn$ covariance matrix; its estimation requires many more replicates. We leave this CRN issue for future research.

Note: In practice, the definition of a 'replicate' may be ambiguous in steady-state simulations. In such simulations, practitioners often make a single 'long' run, and partition this run into *m* subruns to compute the estimated covariances through (5). In both steady-state simulations and terminating simulations, the number of required replicates may exceed the minimum value required for a non-singular estimated covariance matrix—in case of a low signal/noise $E(w)/\sqrt{\text{var}(w)}$ in the simulation at hand. In such a case, additional replicates are required to estimate the gradients with acceptable accuracy. We shall return to this issue in the synthetic example and the inventory simulation.

We use these estimated (co)variances of the simulation outputs—defined in (4)—to estimate the covariance matrix of the regression parameters estimated through (3):

$$\hat{\boldsymbol{cov}}(\hat{\boldsymbol{\beta}}_{h'},\ \hat{\boldsymbol{\beta}}_{h''}) = \hat{\boldsymbol{cov}}_{replicates}(w_{h'}, w_{h''}) \otimes (\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})^{-1} \text{ with } h', h'' = 0, ..., \quad z-1 \tag{7}$$

where $\hat{\boldsymbol{cov}}_{replicates}(w_{h'}, w_{h''})$ is the $z \times z$ matrix defined in (5); $(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})^{-1}$ is a $q \times q$ matrix following from the experimental design $\tilde{\boldsymbol{D}}$ defined below (3) and the first-order or second-order polynomial fitted locally; $\otimes$ denotes the Kronecker product, so $\hat{\boldsymbol{cov}}(\hat{\boldsymbol{\beta}}_{h'},\ \hat{\boldsymbol{\beta}}_{h''})$ is a $zq \times zq$ matrix formed from (4) by multiplying each of its elements by the entire matrix $(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})^{-1}$ (also see Porta Nova and Wilson 1989).

We discuss the following three cases of (7), assuming a first-order polynomial approximation for illustration purposes.

(i) The first (*k* + 1) elements on the main diagonal of $\hat{\boldsymbol{cov}}(\hat{\boldsymbol{\beta}}_{h'},\ \hat{\boldsymbol{\beta}}_{h''})$ are the estimated variances of the estimated main effects—plus the dummy factor corresponding with the intercept—on the goal response w_0; the next (*k* + 1) elements are the estimated variances of the estimated factor effects on response w_1; and so on.

(ii) If $h' = h''$, then (7) concerns the block diagonal. Now (7) estimates the $(k+1)\times(k+1)$ (co)variances between the estimated effects of different factors on a specific output, $w_{h'}$. These effects are correlated—unless the design is orthogonal so $(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})^{-1}$ is diagonal.

(iii) If $h' \neq h''$, then (7) estimates the $(k+1)\times(k+1)$ covariances between the estimated effects on the two simulation outputs $w_{h'}$ and $w_{h''}$. (If the design matrix is orthogonal, then a specific factor still has correlated estimated effects on different responses—because these responses are correlated whenever they are generated by the same scenario. For example, in Section 5, we shall use an orthogonal design for k = 2 factors to generate z = 3 responses per scenario. The estimated effects of the first factor—denoted by the subscript 1— on the first two responses—denoted by the subscripts $h' = 0$ (goal) and $h'' = 1$ (constraint 1) have an estimated covariance $\widehat{\text{cov}}(\hat{\beta}_{0;1}, \hat{\beta}_{1;1})$ equal to $\hat{\sigma}_{0;1}/4$.

4. Testing the KKT conditions

The main goal of this paper is to derive a small-sample procedure to test whether the KKT conditions hold for the ‘current’ solution of problem (1); examples of such a solution are the points labeled A through D in Figure 1. We therefore test whether the following *three null-hypotheses* hold.

(i) First we test whether the current solution is feasible and whether at least one constraint is *binding*. Therefore we compare the z –1 simulation responses w_h (h = 1, …, z –1) with their bounds a_h defined in (1). We test the *center* point of the current local area, because that point is more representative than the (extreme) scenarios of the R-3 design or the non-center points of the CCD; moreover, we avoid the problem of multiple tests (which we could have solved through Bonferroni’s inequality—at the expensive of conservative test results). So we test the following null-hypothesis, which implies zero slack for constraint h:

$$\mathrm{H}_0^{(1)}: \mathrm{E}(w_h(\tilde{\boldsymbol{d}} = \boldsymbol{0})) = a_h \quad (h = 1, \ldots, z-1) \qquad (8)$$

where we use an equality sign instead of the $\geq$ sign in (1)—for reasons discussed at the end of subsection 4.1.

(ii) We replace all deterministic quantities in the original KKT conditions (2) by their (random) estimators; i.e., we test

$$\mathrm{H}_0^{(2)} : E(\hat{\boldsymbol{\beta}}_{-0;\,0}) = E(\hat{\boldsymbol{B}}_{-0;\,J}\hat{\boldsymbol{\lambda}})\,. \tag{9}$$

(iii) We test that $\hat{\boldsymbol{\lambda}}$, the *Lagrange* multipliers estimated in (9), satisfy

$$\mathrm{H}_0^{(3)} : \mathrm{E}(\hat{\boldsymbol{\lambda}}) \geq 0 \tag{10}$$

as discussed below (2).

Next we shall discuss how we test these three hypotheses sequentially.

4.1 Student t test for binding constraints

To test the hypothesis in (8), we use the classic Student test:

$$t_{m-1} = \frac{w_h(\tilde{\boldsymbol{d}} = \boldsymbol{0}) - a_h}{\hat{\sigma}_h / \sqrt{m}} \tag{11}$$

where both the numerator and the denominator use the *m* replicated simulation outputs at the center point (so $\hat{\sigma}_h$ is the 'pure error' standard deviation following from equation 5).

To save simulation runs, a local experiment should start at its center point, including replicates. If it turns out that either no constraint is binding or at least one constraint is violated, then the other hypotheses need not be tested so the remainder of the design is not simulated.

We might replace the *t* statistic defined in (11) by an *F* statistic using $t^2_{m-1} = F_{1;\,m-1}$. However, the *t* statistic enables us to use two different values for the type-I errors corresponding with the two tails of the Student distribution, so we may obtain the following three different results:

(i) When the *t* statistic in (11) gives a *significant positive* value, we conclude that the constraint for output *h* is not binding. If we find that none of the constraints is binding, we conclude that the optimal solution is not yet found; i.e., we assume that at the optimum at least one constraint is binding. In this case, the current local area gives feasible solutions, and the search for better solutions continues—applying one of the simulation-based optimization methods mentioned in Section 1.

(ii) When we find a *significant negative* value, we conclude that the current local area does not give feasible solutions; i.e., the optimal solution is not yet found. The search should back up into the feasible area.

(iii) When we find a *non-significant* value, we conclude that the current local area gives feasible solutions, and that the constraint for output *h* is binding. We include the gradient of this response in $\boldsymbol{B}_J$ defined below (2). And we proceed to test whether the optimal solution is now found—as follows.

4.2 Bootstrap test of KKT conditions

To test the two related hypotheses in (9) and (10), we propose *bootstrapping*. There are two bootstrap types (Efron and Tibshirani 1993):

(i) Parametric: this bootstrap type assumes (for example) *normally* distributed observations.

(ii) Distribution-free: this type does not assume (say) normality—instead it resamples the original data.

We cannot apply (ii), because all points—except for the center point—have a single simulation output so resampling would always give the same observation per scenario. Therefore, we apply (i); i.e., we estimate the parameters of the multivariate normal distribution that plays a role (see equation 14 below). These estimated parameters are (indirectly) computed from the simulation I/O data—also see Figure 2—so the bootstrap is called *data-driven*. (Moreover, our normality assumption would simplify comparison with the asymptotic tests discussed in Section 1—if readers wish to do so.)

Insert Figure 2: I/O of three models: simulation, regression, bootstrap

This figure illustrates that we treat the simulation model as a black box (also see Section 1). Our regression model uses the I/O of the simulation model as input, and estimates the gradients of the goal response (index 0) and the constrained responses, including the binding constraints (index *J*).

The null-hypothesis in (9) states that the goal gradient is a *linear* combination of the gradients of the binding constraints. Obviously, we can always compute such a linear combination through *OLS*:

$$\hat{\hat{\boldsymbol{\beta}}}_{-0;\,0} = \hat{\boldsymbol{B}}_{-0;\,J}(\hat{\boldsymbol{B}}^{T}_{-0;\,J}\hat{\boldsymbol{B}}_{-0;\,J})^{-1}\hat{\boldsymbol{B}}^{T}_{-0;\,J}\hat{\boldsymbol{\beta}}_{-0;\,0} = \hat{\boldsymbol{B}}_{-0;\,J}\hat{\boldsymbol{\lambda}} \qquad (12)$$

where in (3) (the formula for the OLS estimator applied to the simulation I/O data) we replace the deterministic explanatory variable $\boldsymbol{X}$ by *the random* explanatory variable $\hat{\boldsymbol{B}}_{-0;\,J}$, which makes (12) a non-linear function of the multivariate Gaussian variable $vec(\hat{\boldsymbol{\beta}}^{*}_{-0;\,0}, \hat{\boldsymbol{B}}^{*}_{-0;\,J})$; also see (14) below. It is well known that non-linear statistics can be handled through bootstrapping. Altogether, (12) uses the following symbols:

$\hat{\boldsymbol{B}}_{-0;\,J}$: $k \times J$ matrix of estimated gradients of the *J* binding constraints (each gradient has *k* components because there are *k* inputs; gradients follow from the simulation I/O data $(\boldsymbol{X}, \boldsymbol{w})$ via (3); all *z* – 1 constraints are tested in the preceding subsection;

$\hat{\boldsymbol{\beta}}_{-0;\,0}$: goal gradient; see (3) with $h' = 0$;

$\hat{\hat{\boldsymbol{\beta}}}_{-0;\,0}$: OLS estimator of $\hat{\boldsymbol{\beta}}_{-0;\,0}$ as a linear combination of the gradients of the binding constraints;

$\hat{\boldsymbol{\lambda}}$: estimate of the Lagrange multipliers $\boldsymbol{\lambda}$ of the KKT conditions in (2).

Figure 3 illustrates that the OLS estimator $\hat{\hat{\boldsymbol{\beta}}}_{-0;\,0}$ projects $\hat{\boldsymbol{\beta}}_{-0;\,0}$ onto the subspace formed by $\hat{\boldsymbol{B}}_{-0;\,J}$; part A corresponds with the optimal point A in Figure 1, whereas part B corresponds with (say) point B in that figure (the remaining vectors in Figure 3 will be discussed below).

Insert Figure 3: Example of gradients of goal and binding constraint

A classic statistic to measure the *accuracy* of the linear model is the *k*-dimensional vector of *residuals*

$$\hat{\boldsymbol{e}} = \boldsymbol{e}(\hat{\hat{\boldsymbol{\beta}}}_{-0;\,0}) = \hat{\boldsymbol{\beta}}_{-0;\,0} - \hat{\hat{\boldsymbol{\beta}}}_{-0;\,0} \qquad . \tag{13}$$

These residuals $\boldsymbol{e}(\hat{\hat{\boldsymbol{\beta}}}_{-0;\,0})$—denoted by $\hat{\boldsymbol{e}}$ in Figure 3—should not be confused with the fitting errors (say) $\boldsymbol{e}(\hat{\boldsymbol{y}})$ when estimating the simulation output $\boldsymbol{w}$ through $\hat{\boldsymbol{y}}$ using either a first-order or a second-order polynomial; see the numerator in (4).

The question now is: what is an *acceptable* value for the residuals $\boldsymbol{e}(\hat{\hat{\boldsymbol{\beta}}}_{-0;\,0})$ defined in (13), accounting for the randomness in the simulation output—which determines the randomness in the estimated gradients (again see Figure 3)?

To answer this question, we ‘simulate’ gradient values that agree with the observed randomness—quantified through the estimated covariance matrix of the estimated gradients in (7). To generate these values, we sample—via the Monte Carlo method—from the relevant distributions; i.e., we apply *parametric bootstrapping*. This bootstrap procedure consists of the following steps, where we use the standard notation for bootstrapped (sampled, simulated) values, namely the superscript *.

Step 1: We sample the bootstrap values (say) $vec(\hat{\boldsymbol{\beta}}^{*}_{-0;\,0}, \hat{\boldsymbol{B}}^{*}_{-0;\,J})$:

$$vec(\hat{\beta}^{*}_{-0;\,0},\, \hat{\boldsymbol{B}}^{*}_{-0;\,J}) \in N(vec(\hat{\beta}_{-0;\,0},\, \hat{\boldsymbol{B}}_{-0;\,J}),\, \boldsymbol{c\hat{o}v}(vec(\hat{\beta}_{-0;\,0},\, \hat{\boldsymbol{B}}_{-0;\,J}))) \qquad (14)$$

where we define the parameters of this multivariate normal distribution as follows:

$vec(\hat{\beta}_{-0;\,0},\, \hat{\boldsymbol{B}}_{-0;\,J})$: vector with $k + kJ$ elements formed by 'stapling' (stacking) the k elements of the goal gradient, followed by the J vectors of the corresponding $k \times J$ matrix $\hat{\boldsymbol{B}}_{-0;\,J}$ (also see equation 12);

$\boldsymbol{c\hat{o}v}(vec(\hat{\beta}_{-0;\,0},\, \hat{\boldsymbol{B}}_{J}))$: $(k + kJ) \times (k + kJ)$ matrix of estimated (co)variances of the estimated gradients of the goal response and the binding constraints; these (co)variances are computed from (7).

Figure 3 shows one bootstrapped value for $\hat{\beta}^{*}_{-0;\,0}$ and $\hat{\boldsymbol{B}}^{*}_{-0;\,J}$ (besides the original values), for the points A and B in Figure 1.

Step 2: We compute the OLS estimate of the bootstrapped goal gradient, using the bootstrapped gradients of the binding constraints as explanatory variables; i.e. we use (12) with $\hat{\boldsymbol{\beta}}_{-0;\,0}$ replaced by $\hat{\boldsymbol{\beta}}^{*}_{-0;\,0}$ and $\hat{\boldsymbol{B}}_{-0;\,J}$ by $\hat{\boldsymbol{B}}^{*}_{-0;\,J}$ so (12) results in $\hat{\hat{\boldsymbol{\beta}}}^{*}_{-0;\,0}$ and $\hat{\boldsymbol{\lambda}}^{*}$.

We update a variable (say) c^{*} that counts the number of times any of the J bootstrapped Lagrange multipliers $\hat{\boldsymbol{\lambda}}^{*}$ is *negative* (after we have executed our bootstrap procedure—say—1000 times, we test whether this counter is so big that we should reject the hypothesis in equation 10; see below).

Using $\hat{\hat{\boldsymbol{\beta}}}^{*}_{-0;\,0}$ (OLS estimate of bootstrapped goal gradient), we compute the bootstrapped error through (13) with $\hat{\boldsymbol{\beta}}_{-0;\,0}$ replaced by $\hat{\boldsymbol{\beta}}^{*}_{-0;\,0}$ and $\hat{\hat{\boldsymbol{\beta}}}_{-0;\,0}$ by $\hat{\hat{\boldsymbol{\beta}}}^{*}_{-0;\,0}$ so (13) results in the bootstrapped residuals, $\boldsymbol{e}(\hat{\hat{\boldsymbol{\beta}}}^{*}_{-0;\,0}) \equiv \hat{\boldsymbol{e}}^{*}$; also see Figure 3.

We emphasize that $\hat{\boldsymbol{B}}^{*}_{J}$ may be a *square* matrix; for example Angűn et al. (2002) have two binding constraints so $\hat{\boldsymbol{B}}^{*}_{J}$ has two columns, and two inputs so $\hat{\boldsymbol{B}}^{*}_{J}$ is two by two. A square non-singular matrix $\hat{\boldsymbol{B}}^{*}_{J}$ implies that $\hat{\boldsymbol{e}}^{*} = \boldsymbol{0}$ and the projection of $\hat{\boldsymbol{\beta}}^{*}_{-0;\,0}$ is not onto a proper subspace (i.e., a square matrix $\hat{\boldsymbol{B}}^{*}_{J}$ implies $R^2 = 1$ where R^2 denotes the coefficient of determination). In other words, a square non-singular $\hat{\boldsymbol{B}}_J^*$ implies that the random KKT problem reduces to a deterministic problem; for solving such problems we refer to the literature on *deterministic* nonlinear programming, such as Gill et al. (2000). To avoid such 'degeneration' in our numerical example (Section 5), we follow Angűn et al. (2004) and change the parameters in Angűn et al. (2002) such that there is a single binding constraint at the optimum—involving two inputs. (Kleijnen (1993) presents a case study with *k* = 14 inputs that control a decision support system for production planning by a Dutch steel tube manufacturer, which has *z* = 2 outputs so in this practical example $\hat{\boldsymbol{B}}_J^*$ is indeed not a square matrix.)

Step 3: We repeat steps 1 and 2 (say) *R* times (for example, *R* = 1000)—*R* is known as the *bootstrap sample size*. This gives *R* observations on $\boldsymbol{e}(\hat{\hat{\boldsymbol{\beta}}}^*_{-0;\,0}) \equiv \hat{\boldsymbol{e}}^*$, denoted as $\hat{\boldsymbol{e}}_r^*$ (*r* = 1, …., *R*). In addition, step 2 gives *c*—which counts the number of negative Lagrange multipliers $\hat{\boldsymbol{\lambda}}^*$. We point out that it is computationally efficient to replace *R* = 1000 by *R* = 999; see Kleijnen, Cheng, and Bettonvil (2001).

Note: This step's computational time is negligible compared with the computer time needed for the generation of the expensive simulation output $w_{h'}$, used in Section 3 (see again Figure 3.)

Step 4: From Step 3 we compute (say) $\hat{F}_j^*$, the *EDF* per input *j* of the bootstrapped residual; i.e., we sort the results from Step 3 per input, which results in the order statistics $\hat{e}^*_{j;(r)}$ (*j* = 1, …, *k* ; *r* = 1, …, *R*) where the subscript (.) is the standard symbol for order statistics. Besides, we compute the fraction c^* / R of negative Lagrange multipliers.

Step 5: We estimated the two-sided $1-\alpha$ *bootstrap confidence interval per input* from $\hat{e}^*_{j;(\lfloor R\alpha/2 \rfloor)}$, which denotes the lower $\alpha/(2k)$ *quantile* of the EDF computed in step 4 where the bottom or floor function $\lfloor . \rfloor$ implies that we (rather arbitrarily) round to the next integer. (Besides this simple confidence interval, Efron and Tibshirani (1963) and Hall (1987) present several alternatives.)

We reject $\mathrm{H}_0^{(2)}$ defined in (9) if *any* of these *k* confidence interval does not cover zero, we apply *Bonferroni's inequality* to test whether the residuals deviate significantly from zero. We expect that in Figure 3 we tend to 'accept' this null-hypothesis in the optimal point A, whereas in point B we tend to reject.

Note: Bonferroni's inequality provides simple but conservative tests. Alternatively, we could use Tukey's depth in Step 5 to compute a confidence region for all inputs simultaneously; see Yeh and Singh (1997). This, however, requires much more computational effort. In particular, computing Tukey's depth for dimensions higher than 3 (in our case, when the number of inputs k is larger than 3) is still a hard computational problem (see Rousseeuw and Struyf, 1998).

We reject the other null-hypothesis, $\mathrm{H}_0^{(3)}$ defined in (10) if c^* / R—the fraction of *negative* bootstrapped Lagrange multipliers also computed in Step 4—is 'significantly' large. We point out that a Lagrange multiplier that is only 'slightly' larger than zero, has 'nearly' 50% probability of generating negative values if its distribution is symmetric. Therefore we use the binomial distribution to test whether the fraction c^* / R is significantly larger than 50%. We approximate this distribution through the normal distribution with mean 0.50 and variance $(0.50\times0.50)/R$. We expect that in point D of Figure 1 we reject this null-hypothesis, whereas in points A, B, and C we do not.

Note: If we ignored the *random* character of the estimated gradients of the binding constraints, then an alternative test—assuming normally distributed simulation outputs—would be the classic *F*-test (see any textbook on linear regression analysis). The latter test is an exact, small-sample test, comparing

(i) the Sum of Squared Residuals, $SSR = \boldsymbol{e}'(\hat{\boldsymbol{\beta}}_{-0;\,0})\boldsymbol{e}(\hat{\boldsymbol{\beta}}_{-0;\,0})$, of the so-called *full* model;

(ii) the SSR of the *reduced* model that eliminates as explanatory variables all those binding constraints that have *negative* $\boldsymbol{\lambda}$.

5. Synthetic example

To test our statistical procedure, we wish to *guarantee that all its assumptions hold* (in future research, we may test the robustness of our procedure). The two main assumptions are

(i) the simulation outputs ***w*** are multivariate *normal*, and

(ii) the polynomials give adequate fit to the true simulation I/O functions.

Sub (i): In the inventory simulation of the next section, we should make the simulation runs ‘long enough’ to obtain normally distributed responses. Unfortunately, the runs might then be extremely long, so we might need much computer time. Moreover, such runs do not *guarantee* normality: when exactly does asymptotic normality hold for a time series average?

Sub (ii): We know that practical simulation models (such as queueing and inventory simulations) imply *imperfect* fit of first-order and second-order polynomials; such low-order polynomials may be ‘adequate’ if the local area is small ‘enough’—given the magnitude of the noise.

Therefore, we use the same synthetic example as the one in Angün et al. (2004); see again Figure 1. We assume the following true I/O functions: the outputs ***w*** are multivariate normal, with means such that $\mathrm{E}\left(w_h(\boldsymbol{d})\right)\,(h=0,...,z-1)$ are *second-order polynomials* in the two inputs ***d***; the covariance matrix follows below. We select the coefficients of these polynomials such that only one of the two constraints is binding at the true optimum (also see the discussion on square matrices at the end of step 2 in Subsection 4.2). Given these assumptions, we must select specific values for these coefficients. We select these values rather arbitrary, but we do not try to select values that favor the performance of our procedure (actually, we would not know how to select favorable values). Our choices imply that the general problem (1) reduces to

$$\begin{array}{ll} \text{Minimize} & \mathrm{E}\left((d_1-8)^2+(d_2+8)^2+e_0\right) \\ \text{subject to} & \mathrm{E}\left((d_1-3)^2+d_2^2+d_1\,d_2+e_1\right)\leq 4 \\ & \mathrm{E}\left(d_1^2+3(d_2+1.061)^2+e_2\right)\leq 9 \end{array} \qquad (15)$$

where all additive noises *e* are multivariate normal with zero means.

Obviously, (15) implies that the unconstrained minimum would occur at the input combination $\boldsymbol{d} = (8, -8)^T$. It is easy to derive analytically that the constrained minimum occurs at (approximately) $\boldsymbol{d}^o = (2.5328, -1.9892)^T$; see point A in Figure 1.

To generate data for the example in (15), we must select values for $\boldsymbol{cov}(w_{h'}, w_{h''})$ = $\boldsymbol{cov}(e_{h'}, e_{h''})$, which characterizes the noise. Unlike Angűn et al. (2004), we assume no replicates except for a few replicates at the local center. But this assumption implies that we cannot increase the number of replicates to restrict the noise (therefore we do not select the values that Angűn et al. select). Our $\boldsymbol{cov}(w_{h'}, w_{h''})$ determines the *signal/noise* ratio, $\beta / \sqrt{\text{var}(\hat{\beta})}$ —once we have selected the range of the local area; also see the (co)variance formula (7). Two conflicting arguments apply—one mathematical and one statistical (also see Safizadeh 2002):

(i) the smaller the range of the local area, the better the local low-order approximation (Taylor series argument);

(ii) the larger this range, the higher the signal/noise ratio; i.e., the smaller the noise, $\text{var}(\hat{\beta})$.

Inspired by Angűn et al. (2003), we start with the following standard deviations for the simulation responses in all local areas: $\sigma_0 = 1$, $\sigma_1 = 0.15$, $\sigma_2 = 0.4$, and correlations (say) $\rho_{0;1}$= 0.6 $\rho_{0;2}$ = 0.3, and $\rho_{1;2}$ = -0.1. We select the size of the local area rather arbitrarily, after some trial-and-error (also see the discussion of Table 1 below). These choices turn out to give reasonable signal/noise values. In practice, too much signal reduces the problem type defined in (1) to a deterministic problem; too little signal implies that the analysts had better thrown a coin—rather than spend much time on developing a simulation model. (In deterministic optimization, the users also select the size of the so-called 'trust region' subjectively; see Conn, Gould, and Toint 2000.)

To estimate our procedure's *power function*, we apply our procedure in four local areas, each with a center point corresponding with the four points A through D in Figure 1:

(A) The approximately optimal point (2.53, -1.99)

At this point, our test procedure should reject the null-hypothesis (9) and (10) respectively, with probability only α (type-I error rate). We select $\alpha = 0.10$, as Angűn et al. (2004) do; the observed value may be denoted by $\hat{\alpha}$; it is binomially distributed. Note that our multi-stage procedure fails if it rejects the null-hypothesis stating that the estimated slack of constraint #2 (involving w_2) is zero; see (8).

(B) A point 'near' the optimum, and with the same binding constraint as point A

At this point, our procedure should reject the null-hypothesis (9) or (10), with probability *higher* than α ; i.e., our procedure should show increasing power as the point tested moves away from the true optimum. Our procedure should still 'accept' the null-hypothesis in (8) implying a binding constraint #2.

(C) A point 'far away' from the optimum, and with the same binding constraint

Our procedure should now reject the null-hypothesis (9) and (10), with a probability higher than case B's probability.

(D) A point 'far away' from the optimum, and with a different binding constraint

Our procedure should now reject the null-hypothesis (9) and (10), with a probability higher than the case (B) probability. Our procedure should 'accept' the null-hypothesis of a binding constraint #1 (not #2; see case B).

The classic *design* for the fitting of a second-order polynomial (to the simulation's I/O data) is a CCD (first-order polynomials will be discussed at the end of this section). This example has two simulation inputs so $k = 2$. Hence, the number of parameters in the regression metamodel that approximates the example's I/O function is $q = 6$. So the CCD consists of the two-level full factorial , which has 2^2 factor combinations, augmented with a one-factor-at-a-time design with two values c and $-c$ with $c = \sqrt{2}$ (again see Myers and Montgomery 1995, p. 298), and the center point replicated $m = 4$ times (because of the condition $m \geq z + 1$ with z denoting the number of simulation responses, so in this example $z = 3$; see equation 15). Altogether, the CCD uses $n = 9$ combinations of the $k = 2$ inputs to estimated the $q = 6$ parameters of the second-order polynomial approximation (so the CCD is definitely not saturated).

First, our procedure tests whether the $z - 1 = 2$ constraints are binding at the local center. Next it tests whether a second-order polynomial—based on the CCD—is an adequate approximation. If these two tests are passed, then our procedure tests whether the goal gradient can be adequately approximated as a linear model of the binding constraint (the CCD gives the estimated gradients of the objective simulation response w_0 and the constrained simulation responses w_1 and w_2; because a single constraint is binding, the goal gradient with its *k* = 2 components should be estimated as a linear function of the binding constraint's gradient; this estimate uses OLS and gives *k* = 2 residuals for the components of the goal gradient; see equation 13). If this test is passed, then the Lagrange multipliers are tested for their signs.

To get an accurate estimate of the power of our procedure, we run 1000 *macro-replicates* of our example (i.e., we take 1000 sampled vectors of the simulation output $\boldsymbol{w}$ per input combination, estimate 1000 gradients per response, obtain *R* = 999 bootstrap samples per gradient, etc.). This gives Table 1, which displays experimental results for each of the four locations (labeled A through D) and the following two factors:

(i) *Local area size*: When this area is 'large', the four local points corresponding with the 2^2 design change the center point by 0.1. In the 'small' area, they change the size by 0.01.

(ii) *Noise*: A 'small' noise means that the standard deviations are only 10% of the 'large' standard deviations that were specified above ($\sigma_0 = 1$, $\sigma_1 = 0.15$, $\sigma_2 = 0.4$).

We explain the numbers in Table 1 as follows, starting with point A's upper-left element, and proceeding with the other elements in the same row.

69/1000 = 0.07: Our number of macro-replicates is 1000. The first stage of our procedure uses the Student *t* test defined in (11) to test the null-hypothesis in (8), which states that at least one constraint is binding (we know that constraint 2 is binding). This hypothesis is rejected for 69 macro-replicates. The footnote in Table 1 details that of these 69 macro-replicates, 42 macro-replicates rejected the null-hypothesis because the first constraint was found to be 'inactive' (the slack is positive; we know this is true) and the second constraint was violated (negative slack; we know the slack is zero), and the remaining (69 – 42 =) 27 replicates rejected the null-hypothesis because both constraints are found to be inactive.

The other elements in this row give similar numerical results; i.e., the conservative Bonferroni inequality explains why the observed type-I error rate is slightly smaller than the nominal value, 0.10. The other points (B, C, D) have a row with similar results.

Note: For stages 1 and 2, we obtain identical results for large and small local areas, because we use CRN—which result in perfect correlation coefficient of +1 (in our artificial example, the PRN cannot get out of step, whereas in our inventory simulation the synchronization of the PRN may be problematic

79/931 = 0.08: The number of macro-replicates that remains after stage 1 is (1000 – 69 =) 931. Stage 2 uses the classic lack-of-fit *F* test defined in (6), to test the null-hypothesis asserting the adequacy of the second-order polynomial fitted locally (based on the I/O ‘simulation’ data with input data specified by a CCD). Because our problem has multiple simulation responses, we again use Bonferroni’s inequality, which explains that $\hat{\alpha}$ is lower than the prespecified rate, α = 0.10.

Note: We also applied the multivariate lack-of-fit test of Roy et al. (1971). This test, however, gave too many rejections; for example, we obtain $\hat{\alpha}$ = 0.22. More research would be needed to find out why this happens.

We get similar results for the *F* lack-of-fit test in the other cases: see the elements in the same row as 79/931, and the other points (B, C, D).

106/852 = 0.12: The number of macro-replicates that remains after stage 2 is (931 – 79 =) 852. Stage 3 uses bootstrapping to test whether the estimated goal gradient can be adequately expressed as a linear function of the estimated gradient of the binding constraint. Because our problem has multiple simulation inputs (*k* = 2), we again apply Bonferroni’s inequality. We obtain $\hat{\alpha}$ = 0.12, which is slightly higher than the prespecified 0.10. Two other cases give similar results, but one case give $\hat{\alpha}$ = 0.02.

The sub-optimal points (B, C, D) give good results (good power)—except for the case of a small local area with large noise, but then stage 4 rejects the sign of the Lagrange multipliers (see next paragraph).

0/746 = 0.00: The number of macro-replicates of stage 3 is (852 – 106 =) 746. Stage 4 tests whether the ‘adequate’ linear function of stage 3 has positive bootstrapped Lagrange multipliers $\hat{\lambda}^{*}$. None of these macro-replicates gives a negative multiplier—which is not surprising, given the true gradients at point A in Figure 1—except for the case of a small local area with large noise (in the latter case the linear model for the goal gradient may have any sign for its fitted parameters $\hat{\lambda}^*$).

For the points B and C we obtain results that are very similar to point A. For point D (very far away from the optimum) the Lagrange multipliers often have the wrong sign—which is not surprising given the true gradients at this point. (D has a different simulation response—namely, response 1—resulting in a binding constraint; see the footnote.)

We emphasize that the results of stages 3 and 4 should be interpreted together:

(i) If stage 3 rejects the linear model for the goal gradient, then no macro-replicates are left in stage 4 to test the signs of the Lagrange multipliers.

(ii) If stage 3 accepts the linear model for the goal gradient (in case the noise is large), then stage 4 often rejects the signs of the Lagrange multipliers (the estimated gradients of the goal and the binding constraint show so much noise that they may point in the same direction or not).

Table 1 suggests the following conclusions:

(i) Both the *t* test for the identification of the binding constraints, and the lack-of-fit *F* test perform well—independent of the distance from the optimum, the size of the local area, and the magnitude of the noise.

(ii) The farther away from the optimum, the higher the probability of rejecting the model that expresses the estimated goal gradient as a linear function of the estimated gradient of the binding constraint identified sub (i). That (type-I error) probability is acceptable at the optimum itself. However, cases with a small signal and a large noise often do not reject that linear model; fortunately, these cases often give the wrong (negative) signs for the estimated Lagrange multipliers.

Insert Table 1: Fraction of rejected macro-replicates …

Finally, we investigated the consequences of fitting *first-order polynomials* (instead of second-order polynomials) to the simulation’s I/O data. Such polynomials have only $q = k + 1$ parameters, so they require fewer input combinations to be simulated; i.e. an R-3 design instead of a CCD suffices. In our example with only $k = 2$

simulation inputs, our procedure simulates the 2^2 combinations plus the center point, which is still replicated m = 4 (> r, number of simulation responses) times. Hence, n = 5 and q = 3 so a lack-of-fit F test is possible.

We experiment with the same cases as in Table 1. We obtain results that are similar to the results in Table 1, except for the case of large local area with small noise. In the latter case, the lack-of-fit test rejects the polynomial approximation more often; for example, in point A the lack-of-fit test of the first-order approximation gives $\hat{\alpha}$ = (200/927 =) 0.22, whereas Table 1 shows $\hat{\alpha}$ = (77/908 =) 0.08 for the second-order approximation (which we know is perfect for this artificial example). To save space we do not present further details.

6. Illustration: (*s*, *S*) inventory with a service level constraint

We further illustrate our procedure by applying it to a well-known (random) discrete-event dynamic system (DEDS) simulation, namely the (s, S) inventory system with random lead times and a service level constraint investigated by Bashyam and Fu (1998). So—unlike most authors on inventory models—Bashyam and Fu assume a service constraint (instead of a penalty cost for backorders, which implies unconstrained optimization). Moreover, they allow the supplier's orders to cross in time (they assume Poisson lead times with mean 6). For completeness' sake we add that they assume periodic review.

Bashyam and Fu's goal is to find the optimal reorder level s and order-up-to level S (so equation 1 has k = 2 inputs, d_1 and d_2). Further, w_1 in (1) becomes the *fill rate*; i.e., the fraction of demand directly met from inventory at hand; we focus on a target fill rate of 0.99. Hence, the z = 2 responses are the steady-state fill rate and the steady-state expected costs (namely, order setup plus holding costs; order set-up costs K are 36 and holding costs h are 1, unit cost u are 2). Supplier orders arrive at the beginning of a review period. Review is at the end of the period. Customer's demand occurring during the period is IID, namely exponential with mean 100. The simulation starts with an inventory at S (order up-to level).

Bashyam and Fu define an auxiliary variable $Q = S – s$ to estimate the optimal values for s and S (= $s + Q$); the (re)order quantity, however, is not a fixed quantity but varies with the actual ‘inventory position’, defined as stock on hand, minus customer backorders, plus outstanding supplier orders.

Bashyam and Fu apply perturbation analysis to estimate the gradients of the costs and fill rate with respect to s and Q, and the feasible directions method from nonlinear programming to search for the optimum. We, however, apply regression analysis (in the spirit of RSM) to estimate these gradients, for several combinations of s and Q. We refer to Angűn et al. (2003) for details on how RSM may get to these combinations; the search for these combinations is not the focus of our current research. We try several combinations including Bashyam and Fu’s estimated optimal combination of s and S.

Bashyam and Fu estimate the *true* optimum by means of a brute force search consisting of 30,000 periods simulated, replicated 10 times. Their estimate is s = 1435 and Q = 85 or S = 1520. We take their estimate as the analogue of point A in Figure 1. To save computer time, we simulate only 1000 periods, instead of the 20000 periods that Bashyam and Fu simulate in their sophisticated optimization method (they simulate 30000 periods in their brute force method). We select a local area with a range of 10 for both inputs, s and S. We generate 1000 macro-replicates. This gives that all 1000 macro-replicates result in an inactive service constraint! The estimated cost is 1022.10.

Therefore we repeat our experiment switching to s = 1040 and S = 1065, which are values that Bashyam gave us in private communication. Now our estimated costs are 613.71 (which is lower than the 1022.10 obtained for the preceding combination). The service constraint is violated 107 times; this constraint is inactive 4 times—altogether 111/1000 = 0.11 of the macro-replicates proceeds to fit a second-order polynomial (we skipped the first-order polynomial in this illustration). This polynomial is rejected 52/(1000 – 111) = 52/889 = 0.06, which is acceptable given the conservative Bonferroni's inequality applied to the two polynomials (z = 2 simulation responses). The OLS model expressing the goal gradient as a linear function of the service constraint gradient, gives a rejection rate of 59/((889 – 52) = 59/837 = 0.07. The 'accepted' models have the wrong signs for the Lagrange multipliers in 80/(837 – 59) = 80/778 = 0.10. So we conclude that our procedure has an acceptable probability of accepting this (s, S) combination as being optimal.

We do not search for suboptimal combinations that lie on the fill rate constraint, since such a search seems a project in itself. Moreover, all that this search can give is combinations that should be rejected more frequently than the optimal combination; and the artificial example has already clearly shown that our procedure does have increasing power as the combinations move away from the optimum.

Note: Safizadeh (2002) investigates the *local area size* in RSM, using a similar inventory simulation (but he assumes a shortage cost instead of a fill rate constraint). He experiments with a range of 4, 10, and 20 respectively—for the two inputs, starting with the center point $(175, 175)^{\mathrm{T}}$. He simulates a warm-up period of 30 followed by 5000 time periods—which determine the resulting noise. He recommends small ranges—but he applies CRN, whereas we use independent PRN.

7. Conclusions and further research

The literature on simulation-optimization offers many heuristic search methods. We derived a stopping rule for such methods, supposing that the search has lead to some local area.

The analysts should start with the simulation of the input combination specified by the *center* of the local area. Replicating this combination $m = r + 1$ times, they can use the t statistic to test whether any constraint is binding (stage 1 of our procedure).

If a binding constraint is found, then the analysts may start with a *first-order polynomial approximation* of the simulation's I/O function in the local area and use the F lack-of-fit test (stage 2 of our procedure).

If this test does not reject this approximation, then the analysts may estimate the gradients from these r polynomials; otherwise, the analysts can switch to a second-order polynomial approximation, and augment the R-3 design to a CCD. Using either a first-order or a second-order polynomial approximation, the analysts use the corresponding estimated gradients to bootstrap the gradients of the goal function and the binding constraint(s). This bootstrap enables testing whether the goal gradients can be adequately approximated by a linear function—estimated through OLS—of the binding constraint gradients (stage 3 of our procedure test whether the bootstrapped residuals of this OLS model are zero).

Finally, if this approximation is adequate, then the Lagrange multipliers—estimated through OLS—should be non-negative (tested in stage 4).

In future research, we may apply our procedure to *realistic* simulation models; for example, the call center model in Kelton, Sadowski, and Sadowski (2002), which is so complicated that it is eliminated in the latest edition, Kelton, Sadowski, and Sturrock (2004).

Further, CRN might be applied to improve the accuracy of the estimated gradients. Unfortunately, CRN requires many more replicates to estimate the covariance matrix for the simulation responses; also see Kleijnen (1992).

We also tried to measure the accuracy of the linear model that expresses the goal gradients in the binding constraints' gradients through the SSR (also see the Note at the end of Section 4). If this model is adequate, then this SSR is still higher than zero (whereas the expected residuals are zero). Mysteriously, our experiments with the artificial example gave much to high estimated type-I error rates. This deserves further research.

In the various stages of our procedure, we use Bonferroni's inequality, which provides simple but conservative tests. Future work could focus on using the notion of data depth in stage 3 of our method.

Finally, we may test *second-order* optimality conditions (besides the first-order KKT conditions).

References

Angűn, E., D. den Hertog , G. Gürkan, and J.P.C. Kleijnen (2003), Response surface methodology with stochastic constrains for expensive simulation. Working Paper (download from http://center.kub.nl/staff/kleijnen/papers.html)

Angűn, E., D. den Hertog , G. Gűrkan, and J.P.C. Kleijnen (2002), Response surface methodology revisited. *Proceedings of the 2002 Winter Simulation Conference* (edited by E. Yücesan, C.H. Chen, J.L. Snowdon and J.M. Charnes), pp. 377-383

Angűn, E., and J.P.C. Kleijnen (2004), An asymptotic stopping rule for Response Surface Methodology with stochastic constraints. Working Paper (download from http://center.kub.nl/staff/kleijnen/papers.html)

Bashyam, S. and M.C. Fu (1998), Optimization of (s, S) inventory systems with random lead times and a service level constraint. *Management Science*, 44, no, 12, Part 2, p. 243-256

Conn, A.R., N. Gould, and Ph. L. Toint (2000), *Trust-region methods*. SIAM, Philadelphia

Dykstra, R.L. (1970), Establishing the positive definiteness of the sample covariance matrix. *The Annals of Mathematical Statistics*, 41, no. 6, pp 2153-2154

Eaton, M.L.(1983), *Multivariate statistics, a vector space approach*. Wiley, New York

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

Fu, M.C. (2002), Optimization for simulation: theory vs. practice. *INFORMS Journal on Computing*, 14, pp. 192-215

Gill, P. E., W. Murray, M. H. Wright (2000). *Practical optimization, 12th edition*. Academic Press, London

Hall, P. (1987), On the bootstrap and likelihood-based confidence region. *Biometrika*, 74, no. 3, pp. 481-493

Joshi, S., H.D. Sherali, and J.D. Tew (1998), An enhanced response surface methodology (RSM) algorithm using gradient deflection and second-order search strategies. *Computers and Operations Research*, 25, no. 7/8, pp. 531-541

Karaesman, I and G. van Ryzin (2004), Overbooking with substitutable inventory classes, *Operations Research*, 52, no. 1, pp. 83-104

Kelton, W.D., R.P. Sadowski, and D.A. Sadowski (2002), *Simulation with Arena; second edition.* Mc Graw-Hill, Boston

Kelton, W.D., R.P. Sadowski, and D.T. Sturrock (2004), *Simulation with Arena; third edition.* Mc Graw-Hill, Boston

Khuri, A.I. (1996), Multiresponse surface methodology. *Handbook of Statistics, vol. 13*, edited by S. Ghosh and C.R. Rao, Elsevier, Amsterdam

Kleijnen, J.P.C. (1993), Simulation and optimization in production planning: a case study. *Decision Support Systems*, 9, pp. 269-280

Kleijnen, J.P.C. (1992), Regression metamodels for simulation with common random numbers: comparison of validation tests and confidence intervals. *Management Science*, 38, no. 8, pp. 1164-1185

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

Kleijnen, J.P.C., R.C.H. Cheng, and B. Bettonvil (2001), Validation of trace-driven simulation models: bootstrapped tests. *Management Science*, 47, no. 11, pp. 1533-1538

Myers, R. H. and D. C. Montgomery (2002), *Response Surface Methodology: Process and product optimization using designed experiments; second edition*. John Wiley & Sons, New York

Porta Nova, A.M. and J.R. Wilson (1989), Estimation of multiresponse simulation metamodels using control variates. *Management Science*, 35, no. 11, pp. 1316-1333

Rao, C. R. (1967), Least squares theory using an estimated dispersion matrix and its application to measurement of signals. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, I,* pp. 355-372

Rousseeuw, P.J. and Struyf, A. (1998), Computing location depth and regression depth in higher dimensions, *Statistics and Computing*, **8**, 193-203.

Roy, S.N., R. Gnanadesikan, and J.N. Srivastava (1971), *Analysis and design of certain quantitative multiresponse experiments*. Pergamon Press, Oxford

Ruud, P. A. (2000), *An introduction to classical econometric theory*. Oxford University Press, New York

Safizadeh, M.H. (2002), Minimizing the bias and variance of the gradient estimate in RSM simulation studies. *European Journal Operational Research*, 136, no. 1, pp. 121-135

Shao, J and D. Tu (1995), *The jackknife and bootstrap*. Springer-Verlag, New York

Shapiro, A. (2000), Statistical inference of stochastic optimization problems. *Probabilistic constrained optimization: theory and applications*, edited by S.P. Uryasev, Kluwer, pp. 91-116

Shapiro, A. and T. Homem-de-Mello (1998), A simulation-based approach to two-stage stochastic programming with recourse. *Mathematical Programming*, 81, pp. 301-325

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

Yeh, A.B. and K. Singh (1997), Balanced confidence regions based on Tukey's depth and the bootstrap. *Journal of the Royal Statistical Society, Series B (Methodological)*, 59, no. 3, pp. 639-652

Table 1: Fraction of rejected macro-replicates

in four local areas centered around A, B, C, and D in Figure 1; α = 0.10

A: (2.53,-1.99)	large local region		small local region	
	large noise	small noise	large noise	small noise
binding constraints	69/1000 = 0.07 [1]	92/1000 = 0.09 [2]	69/1000 = 0.07 [1]	92/1000 = 0.09 [2]
polynomial fit	79/931 = 0.08	77/908 = 0.08	79/931 = 0.08	77/908 = 0.08
linear KKT model	106/852 = 0.12	107/831 = 0.13	16/ 852 = 0.02	98/831 = 0.12
positive $\hat{\lambda}^*$	0/7346 = 0.00	0/724 = 0.00	550/836 = 0.66	0/733 = 0.00

[1] 42 times: first constraint inactive, second constraint violated

27 times: both constraints inactive

[2] 26 times: first constraint invalid, second constraint violated

66 times: both constraints inactive

B: (2.00,-2.35	large local region		small local region	
	large noise	small noise	large noise	small noise
binding constraints	67/1000 = 0.07 [1]	101/1000 = 0.10 [2]	67/1000 = 0.07 [1]	101/1000 = 0.10 [2]
polynomial fit	81/933 = 0.09	79/899 = 0.09	81/933 = 0.09	79/899 = 0.09
linear KKT model	232/852 = 0.27	820/820 = 1.00	17/ 852 = 0.02	210/820 = 0.26
positive $\hat{\lambda}^*$	0/620 = 0.00	0/0	541/852 = 0.63	0/610 = 0.00

[1] 40 times: first constraint inactive, second constraint violated

27 times: both constraints inactive

[2] 16 times: first constraint invalid, second constraint violated

85 times: both constraints inactive

C: (3.00,-1.10	large local region		small local region	
	large noise	small noise	large noise	small noise
binding constraints	75/1000 = 0.08 [1]	82/1000 = 0.08 [2]	75/1000 = 0.08 [1]	82/1000 = 0.08 [2]
polynomial fit	73/925 = 0.08	77/918 = 0.08	73/925 = 0.08	77/918 = 0.08
linear KKT model	548/852 = 0.64	841/841 = 1.00	17/ 852 = 0.02	538/841 = 0.64
positive $\hat{\lambda}^*$	4/304 = 0.01	0/0	598/835 = 0.72	2/303 = 0.01

[1] 49 times: first constraint inactive, second constraint violated
26 times: both constraints inactive

[2] 66 times: first constraint invalid, second constraint violated
16 times: both constraints inactive

D: (1.00,-1.00	large local region		small local region	
	large noise	small noise	large noise	small noise
binding constraints	67/1000 = 0.07 [1]	67/1000 = 0.07 [1]	67/1000 = 0.07 [1]	67/1000 = 0.07 [1]
polynomial fit	81/933 = 0.09	81/933 = 0.09	81/933 = 0.09	81/933 = 0.09
linear KKT model	847/852 = 0.99	852/852 = 1.00	33/ 852 = 0.04	847/852 = 0.99
positive $\hat{\lambda}^*$	5/5 = 1.00	0/0	735/819 = 0.90	5/5 = 1.00

[1] 44 times: first constraint violated, second constraint inactive
23 times: both constraints inactive

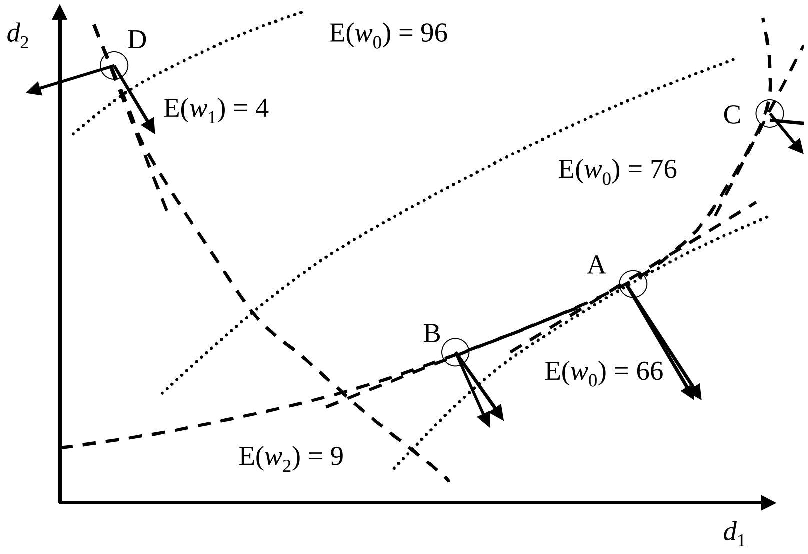

Figure 1: An example of a constrained nonlinear random optimization problem

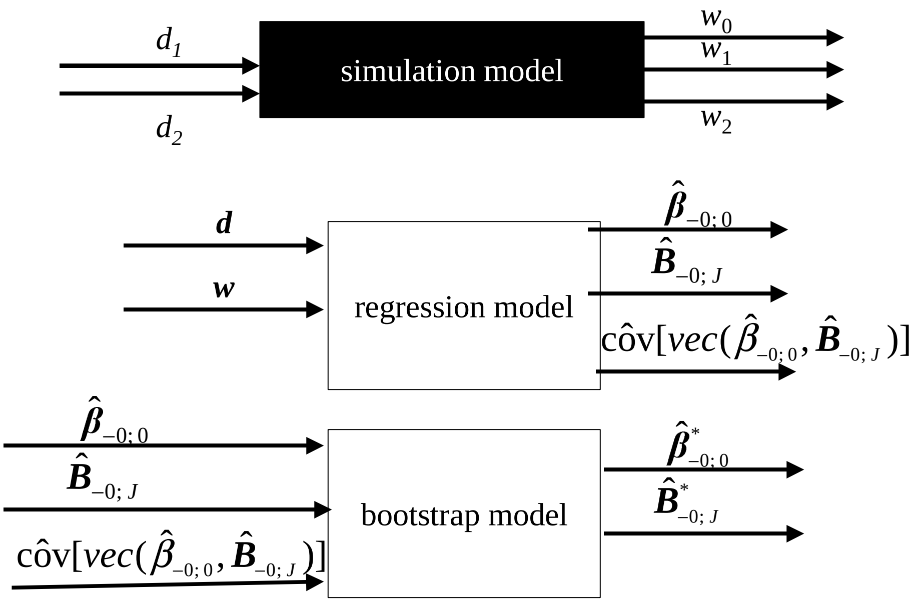
d_1
d_2
simulation model
w_0
w_1
w_2
$\boldsymbol{d}$
$\boldsymbol{w}$
regression model
$\hat{\boldsymbol{\beta}}_{-0;\,0}$
$\hat{\boldsymbol{B}}_{-0;\,J}$
$\widehat{\mathrm{cov}}[vec(\hat{\boldsymbol{\beta}}_{-0;\,0}, \hat{\boldsymbol{B}}_{-0;\,J})]$
$\hat{\boldsymbol{\beta}}_{-0;\,0}$
$\hat{\boldsymbol{B}}_{-0;\,J}$
$\widehat{\mathrm{cov}}[vec(\hat{\boldsymbol{\beta}}_{-0;\,0}, \hat{\boldsymbol{B}}_{-0;\,J})]$
bootstrap model
$\hat{\boldsymbol{\beta}}^{*}_{-0;\,0}$
$\hat{\boldsymbol{B}}^{*}_{-0;\,J}$

Figure 2: I/O of three models: simulation, regression, bootstrap

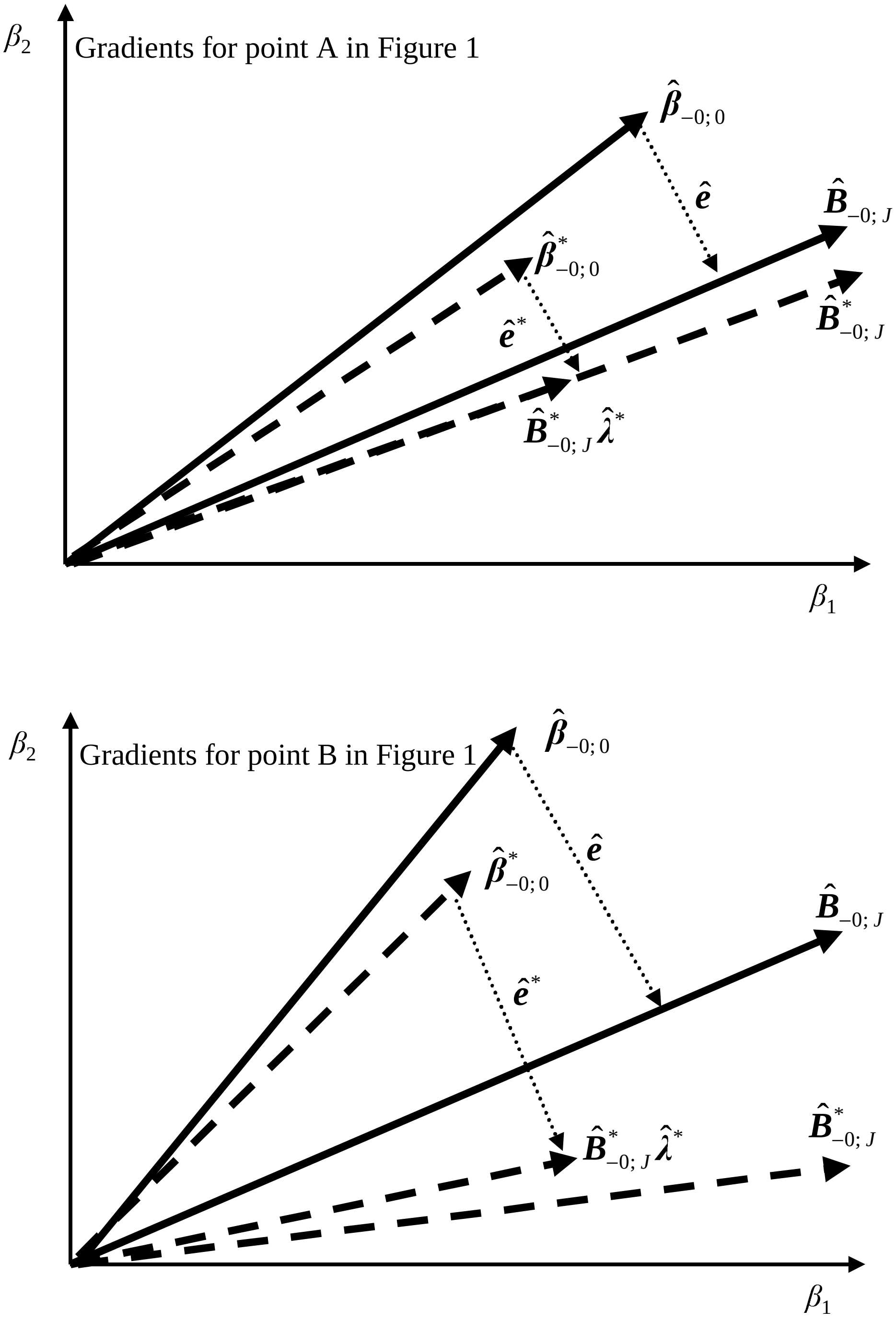
β_2
Gradients for point A in Figure 1
$\hat{\boldsymbol{\beta}}_{-0;\,0}$
$\hat{\boldsymbol{e}}$
$\hat{\boldsymbol{B}}_{-0;\,J}$
$\hat{\boldsymbol{\beta}}^*_{-0;\,0}$
$\hat{\boldsymbol{e}}^*$
$\hat{\boldsymbol{B}}^*_{-0;\,J}$
$\hat{\boldsymbol{B}}^*_{-0;\,J}\hat{\boldsymbol{\lambda}}^*$
β_1
β_2
Gradients for point B in Figure 1
$\hat{\boldsymbol{\beta}}_{-0;\,0}$
$\hat{\boldsymbol{e}}$
$\hat{\boldsymbol{\beta}}^*_{-0;\,0}$
$\hat{\boldsymbol{B}}_{-0;\,J}$
$\hat{\boldsymbol{e}}^*$
$\hat{\boldsymbol{B}}^*_{-0;\,J}\hat{\boldsymbol{\lambda}}^*$
$\hat{\boldsymbol{B}}^*_{-0;\,J}$
β_1

Figure 3: Example of gradients of goal and binding constraint;
part A (respectively B) corresponds with point A (respectively B) in Figure 1;
original and bootstrapped (drawn and dotted lines)