S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*
Copyright © 2006 Elsevier B.V. All rights reserved
DOI: 10.1016/S0927-0507(06)13018-2

Chapter 18

Metamodel-Based Simulation Optimization

Russell R. Barton
The Pennsylvania State University, USA
E-mail: rbarton@psu.edu

Martin Meckesheimer
Boeing Company, USA
E-mail: martin.meckesheimer@boeing.com

Abstract

Simulation models allow the user to understand system performance and assist in behavior prediction, to support system diagnostics and design. Iterative optimization methods are often used in conjunction with engineering simulation models to search for designs with desired properties. These optimization methods can be difficult to employ with a discrete-event simulation, due to the stochastic nature of the response(s) and the potentially extensive run times. A metamodel, or model of the simulation model, simplifies the simulation optimization in two ways: the metamodel response is deterministic rather than stochastic, and the run times are generally much shorter than the original simulation. Metamodels based on first- or second-order polynomials generally provide good fit only locally, and so a series of metamodels are fit as the optimization progresses. Other classes of metamodels can provide good global fit; in these cases one can fit a (global) metamodel once, at the start of the optimization, and use it to find a design that will meet the optimality criteria. Both approaches are discussed in this chapter and illustrated with an example.

1 Introduction

Simulation models provide insight on the behavior of real systems. This insight can be used to improve system performance by ad hoc changes to the system design parameter values, or the simulation model may be analyzed repeatedly to find a set of design parameters that provide the best simulated performance. We define simulation optimization as the latter case: repeated analysis of the simulation model with different values of design parameters, in an attempt to identify best simulated system performance. The design parameters of the real system are set to the 'optimal' parameter values determined by the simulation optimization exercise, rather than in an ad hoc manner based on qualitative insights gained from exercising the simulation model. We will use the following notation to represent the general simulation optimization problem, following Andradóttir (1998):

$$\min_{\theta \in \Theta} f(\theta), \tag{1}$$

where θ is the (possibly vector-valued) design parameter of the system being simulated, and the feasible region $\Theta \in \mathbb{R}^d$ is the set of possible values of θ. The optimization model response function is represented by $f(\theta)$ which is usually the expected value (long-term average) of some simulated system performance measure Y as a function of the design parameter vector θ. That is,

$$f(\theta) = \mathrm{E}\bigl(Y(\theta)\bigr).$$

The form of f is not known. Its value is estimated using n runs of the simulation model under the design scenario specified by θ,

$$\widehat{f(\theta)} = \sum_{i=1}^{n} Y_i \Big/ n, \tag{2}$$

where the dependence of Y on the value of θ has been suppressed. While $f(\theta)$ is deterministic, its estimate is stochastic, since the simulation run time must be finite (so $n < \infty$).

Simulation optimization strategies depend on the nature of f and Θ. When the feasible set of design parameter vector values Θ is a discrete set, appropriate optimization methods include ranking and selection (Chapter 17), random search (Chapter 20) and metaheuristics (Chapter 21). If Θ is continuous and f is differentiable, then gradient-based methods (Chapter 19) or metamodel-based optimization (this chapter) can be used. This structure is shown in Figure 1.

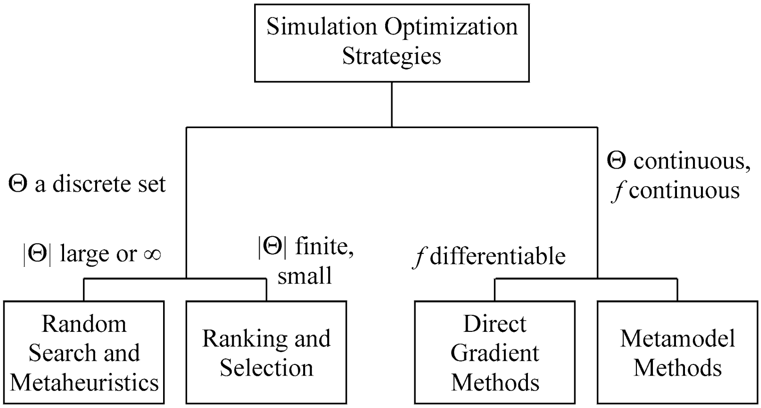


Fig. 1. Simulation optimization strategy depends on the nature of Θ and f.

Table 1.
Simulation optimization strategies (+, advantage; −, disadvantage)

Strategy	Coding modifications to simulation	External 'custom' (non-RSM) metamodel code	External standard statistical (RSM) code	External optimizer code	Efficiency	Provides insight on local response behavior	Provides insight on global response behavior
Random search and metaheuristics				−			
Ranking and selection				−			
Direct gradient methods	−			−	+	+	
Response surface methodology			−			+	
Global metamodel optimization		−		−		+	+

The properties of these general classes of optimization strategies are shown in Table 1. Random search and metaheuristics attempt to select θ values from Θ efficiently. This generally requires a specialized code external to the simulation package. Ranking and selection methods assume that a set of θ alternatives is given, and determine the number of simulation replications and run lengths required for each alternative to give a pre-specified probability of selecting the best or near-best θ. The effectiveness of ranking and selection in an optimization setting depends on the method for choosing candidates. Again, external calculations are necessary to compute run lengths and replications.

If Θ is continuous, other search methods can be employed. Stochastic gradient-based optimization methods such as stochastic approximation can use efficient methods for estimating the gradient of f such as likelihood ratios (these require modification of the simulation code – see Table 1), or less efficient finite-difference approximations. These methods search Θ to find optimum system performance without attempting to provide a global approximation to f, an efficiency advantage (see Section 5.4 in Fu, 1994). The stochastic optimization code is usually external to the simulation code, and can be complex to implement. Simplex search combines features of ranking and selection and gradient optimization, but can fail on stochastic responses with large variation (Tomick et al., 1995). Metamodel-based optimization methods fall in two categories, both of which use an indirect-gradient optimization strategy. It is indirect because the gradient is computed for the metamodel (a deterministic function) rather than for the simulation response. Response surface methodology (RSM) is a metamodel-based optimization method that builds linear or quadratic local approximations to f to be used by a deterministic gradient-based optimization strategy. Old local models are discarded and new ones are fitted at the end of each line search cycle. The global metamodel-based optimization methods build a single global metamodel (usually requiring a much larger set of simulation runs) which is then optimized using a gradient-based strategy.

In some cases metamodel-based optimization can be used with discrete-valued design parameters. If the parameter has a natural integer order, say the number of machines in a work cell, then the continuous approximation can be solved using a metamodel strategy, and the solution parameter value rounded to the nearest (feasible) integer value.

This brief description of simulation optimization serves to place metamodel-based optimization in the overall context of simulation optimization. Fu (2001) provides an overview of simulation optimization methods and the implementations that exist in commercial software.

The remainder of this chapter focuses on metamodel-based optimization strategies. The next section provides an overview of metamodel types and their appropriateness for discrete-event simulation response functions. The following section describes the overall strategy of metamodel-based optimization, and highlights the differences between local and global metamodel approaches. Section 4 describes the RSM approach in more detail and highlights important issues in its use. RSM is illustrated using a network routing design example. Section 5 examines one global metamodel-based approach and applies it to the same optimization case. Section 6 provides a summary and describes how metamodels can be used for robust design, whose objective is to simultaneously optimize the expected value and the standard deviation of the response. Any discussion of metamodel-based optimization necessitates reference to concepts from a variety of fields, including simulation, statistics, response surfaces, and nonlinear optimization. For an introduction to concepts and terminology in these areas, see Banks et al. (2005) and Law and Kelton (2000) for simulation, Box and Draper (1987), Khuri and Cornell (1987), Montgomery (2001), Myers and Montgomery (2002) and Santner et al. (2003) for statistics and response surfaces, and Bertsekas (1999) for nonlinear optimization.

2 Metamodels and simulation

Experimentation with computer simulation models of proposed or existing real systems is often used to make decisions on changes to the system design. Analysts exercise the simulation model because cost, time or other constraints prohibit experimentation with the real system. For the extensive experimentation required for optimization, the simulation models themselves may require excessive computation, and so simpler approximations are often constructed; models of the model, called metamodels (Kleijnen, 1975a, 1975b, 1987) or surrogate models (Yesilyurt and Patera, 1995). These metamodels are usually deterministic approximating functions for f that are inexpensive to compute. Running multiple replications of the simulation to produce $\widehat{f(\theta)}$ is expensive;

running the metamodel once produces the deterministic value $g(\theta)$ which approximates $f(\theta)$ with low computational expense.

The major issues in metamodeling include: (i) the choice of a functional form for g, (ii) the design of experiments, that is, the selection of a set of θ values at which to observe $Y(\theta)$ by running the simulation model, the assignment of random number streams, the length of runs, etc., (iii) fitting g to the simulation response using the experimental data, and (iv) the assessment of the adequacy of the fitted metamodel (confidence intervals, hypothesis tests, lack of fit and other model diagnostics). We will restrict this discussion to the case of a single output performance measure, say total cost. Multiple output measures would each require a separate metamodel.

The functional form for g is typically a linear combination of basis functions from a parametric family. There are choices for parametric families (polynomials, sine functions, piecewise polynomials, etc.) and choices for fitting; that is, choosing the 'best' representation from within a family (via least squares, maximum likelihood, cross-validation, etc.). This section draws from earlier metamodel review papers (Barton, 1992, 1998), with a focus on the most promising metamodel and experiment design strategies for simulation optimization.

2.1 Response surface metamodels

Response surface models were developed over fifty years ago for "the exploration and exploitation" of stochastic response functions (Box and Wilson, 1951; Box, 1954). They are used in conjunction with response surface methodology, the most commonly used approach to metamodel-based simulation optimization. This metamodel family consists of first or second-order polynomial probability models fitted to observed values of Y, the system response. A full second-order response surface model would be

$$Y(\theta) = \beta_0 + \sum_{j=1}^{p} \beta_j \theta_j + \sum_{i=1}^{p} \sum_{k=i}^{p} \beta_{ik} \theta_i \theta_k + \varepsilon, \quad \varepsilon \sim \mathrm{NID}\bigl(0, \sigma^2\bigr), \tag{3}$$

where NID indicates that the deviations have independent (and identical) normal distributions.

Suppose that an experiment has been conducted, with simulation runs at parameter settings $\theta^1, \theta^2, \ldots, \theta^n$ and corresponding observed responses (perhaps averages of replications) of $y^1, \ldots, y^n$. Let y represent the vector of responses. For metamodel prediction, maximum likelihood (equivalently, least squares) estimators for the β_0, β_i, β_{ik} and σ^2 are computed, and used in the prediction equation

$$g(\theta) = \beta_0 + \sum_{j=1}^{p} \beta_j \theta_j + \sum_{i=1}^{p} \sum_{k=i}^{p} \beta_{ik} \theta_i \theta_k. \tag{4}$$

Response surface metamodels can be fit using standard statistical packages. Experiment designs for RSM models and other details of this method are discussed in Section 4.

2.2 Regression spline metamodels

If a linear or quadratic polynomial regression does not provide a good fit, it is natural to think of higher-order polynomial approximations. Any polynomial regression model can be constructed from linear combinations of the functions $\prod_k \theta_{j_k}$, where the index j_k may take the same value more than once. This choice for a basis has drawbacks. The high-order polynomial achieves a good fit by adjusting coefficients to achieve cancellation of large oscillations over most of the range. This reliance on cancellation makes high-order polynomial fits nonrobust. Figure 2 shows a 14th degree polynomial fitted to a sample of deterministic responses at 15 points evenly spaced in the interval $[-5, 5]$ for the response function $f(\theta) = 1/(1 + \theta^2)$. The magnitude of the overshoot at the extremes in the figure would increase if the number of design points and the degree of the polynomial were further increased.

If a linear, quadratic, or cubic approximation to the function is adequate, then polynomial basis functions can be used to construct an effective metamodel. If this is not adequate, the simulation modeler should consider other basis functions from which to build the metamodel. The difficulties with polynomial basis functions are avoided if: (i) they are applied to a small region, and (ii) only low order polynomials are used. This is the motivation for metamodels based on piecewise polynomial basis functions. When continuity restrictions are applied to adjacent pieces, the piecewise polynomials are called splines. The (univariate) metamodel can be written as

$$g(\theta) = \sum_k \beta_k B_k, \tag{5}$$

where the B_k are quadratic or cubic piecewise polynomial basis functions. The basis functions can be described most simply for the univariate case. The domain is divided into intervals $[t_1, t_2), [t_2, t_3), \ldots, [t_{m-1}, t_m)$ whose endpoints

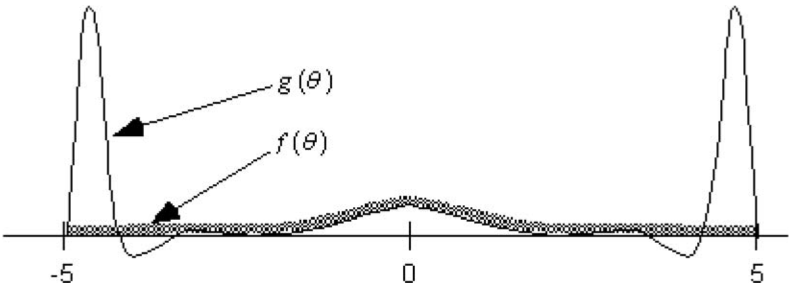


Fig. 2. 14th degree regression polynomial $g(\theta)$ fitted to data from $f(\theta) = 1/(1 + \theta^2)$.

are called knots. Two sets of spline basis functions are commonly used, the truncated power function basis and the B-spline basis (de Boor, 1978). Since most simulation model output functions will not be deterministic, interpolating splines will not be satisfactory. The motivation for smoothing splines is based on an explicit tradeoff between the fit/accuracy of the approximation at known points and smoothness of the resulting metamodel. The fit term is represented as a sum of squared differences of the metamodel and simulation model responses at each of the experimental runs. The smoothness is represented by an integral of the square of some derivative over the region of validity of the metamodel. The relative weight of these objectives is captured by the smoothing parameter, λ: $\lambda = 0$ provides interpolation with no constraint on smoothness. The function that minimizes this quantity will be a spline of order q, which is in C^{q-2} (continuous derivatives up to the $(q-2)$nd derivative) and is a piecewise polynomial with terms up to θ^{q-1}. The knots will occur at points in Θ corresponding to the observed data, θ^i.

An important issue is the selection of the value for the smoothing parameter λ. The value may be chosen by visual examination of the fit, or by minimizing cross-validation (like residual sum of squares), or generalized cross-validation (GCV) (an adjusted residual sum of squares). Eubank (1988) and Craven and Wahba (1979) discuss these approaches.

Smoothing splines are appropriate for simulation metamodels, but the developments have focused on univariate and bivariate functions. The global metamodel example in Section 5 uses the bivariate smoothing spline code of Dierckx (1981, 1993). Unfortunately, the most popular and effective multivariate spline methods are based on interpolating splines, which have little applicability for simulation optimization (Breiman, 1991; Friedman, 1991).

2.3 Spatial correlation (kriging) metamodels

Sacks et al. (1989) and numerous references therein develop a spatial correlation parametric regression modeling approach. The expected smoothness of the function is captured by a spatial correlation function. Spatial correlation models, also called kriging models, have recently become popular for deterministic simulation metamodels (Simpson et al., 1998; Booker et al., 1999). They are more flexible than polynomial models in fitting arbitrary smooth response functions, and seem to be less sensitive than radial basis functions to small changes in the experiment design (Meckesheimer et al., 2002).

Mitchell and Morris (1992) describe the spatial correlation model that is appropriate for (stochastic) simulation responses. The probability model represents the simulation response, Y as

$$Y(\theta) = Z(\theta) + \varepsilon, \tag{6}$$

where ε are independent Gaussian random quantities with mean zero and variance α^2 and Z represents a Gaussian process with mean $\mu(\theta) = \mathrm{E}(Y(\theta))$ and variance σ^2 that exhibits *spatial correlation*,

$$\mathrm{Cov}\big(Z(u), Z(v)\big) = \sigma^2 \mathrm{R}(u, v),$$

where R describes the spatial correlation function. Mitchell and Morris (1992) list four spatial correlation functions, the most commonly used being

$$\mathrm{R}(u, v) = \prod_{j=1}^{p} \exp\big(-\omega_j |u_j - v_j|^2\big), \tag{7}$$

where the index j runs over the dimension of Θ. This Gaussian correlation structure gives an infinitely differentiable metamodel.

Suppose that an experiment has been conducted, with simulation runs at parameter settings $\theta^1, \theta^2, \ldots, \theta^n$ and corresponding observed responses (perhaps averages of replications) of $y^1, \ldots, y^n$. Let y represent the vector of responses. For metamodel prediction, maximum likelihood estimators for the ω_j, μ, σ^2 and α^2 are computed, and used in the prediction equation for $\mathrm{E}(Y)$

$$g(\theta) = \mu + r'(\theta) C^{-1}(y - \mu\mathbf{1}), \tag{8}$$

where $r'(\theta)$ has components $\sigma^2\mathrm{R}(\theta, \theta^i)$, $C_{jk} = \sigma^2\mathrm{R}(\theta^i, \theta^k) + \alpha^2 I(i = k)$, and I is the indicator function. The matrix C depends on θ^i but not on θ. Using γ_i to represent the elements of the matrix–vector product $C^{-1}(y - \mu\mathbf{1})$ makes the form of the basis functions of θ for the spatial correlation metamodel clearer (with μ, γ_i and ω_j as the fitted coefficients).

$$g(\theta) = \mu + \sum_{i=1}^{n} \gamma_i \prod_{j=1}^{p} \exp\big(-\omega_j |\theta_j - \theta_j^i|^2\big).$$

For fitting deterministic simulation models, the spatial correlation model excludes the ε term, and the resulting approximation provides an interpolating fit to the experimental data.

Although discussed in Barton (1992), Mitchell and Morris (1992) and Barton (1998), spatial correlation models have not been applied in the discrete-event simulation context until recently. See Kleijnen (2005) for an overview and Kleijnen and van Beers (2005) for an assessment of robustness of this approach in the presence of heterogeneous variance that often characterizes simulation response functions. The book by Santner et al. (2003) gives a good review of spatial correlation models, and provides the PeRK code for fitting and prediction.

Factorial experiment designs can cause ill-conditioned likelihood functions for spatial correlation metamodels. Orthogonal array, Latin hypercube and orthogonal array-based Latin hypercubes have been shown to be effective (Jin et al., 2000; Meckesheimer et al., 2002). A set of C routines written by Art Owen at Stanford University generate orthogonal arrays, Latin hypercube designs, and orthogonal array based Latin hypercube designs. The routines are available from Statlib at *http://lib.stat.cmu.edu*.

2.4 Radial basis function metamodels

Radial basis functions (RBF) provide an alternative approach to multivariate metamodeling. In an empirical comparison, Franke (1982) found radial basis functions to be superior to thin plate splines, cubic splines, B-splines, and several others for fitting deterministic response functions. Tu and Barton (1997) found them to provide effective metamodels for electronic circuit simulation models, and Shin et al. (2002) applied a radial basis function metamodel to a queueing simulation and cited its potential.

The radial basis function approximation consists of a sum of radially symmetric functions centered at different points in the domain Θ. The original development by Hardy (1971) introduced simple "multiquadric" basis functions $\|\theta - c^k\|$ (where $\|\cdot\|$ denotes Euclidean distance) to give the metamodel form

$$g(\theta) = \sum_{k=1}^{r} \gamma_k \|\theta - c^k\|. \tag{9}$$

The parameters to be chosen are the basis function centers c^k, and the coefficients (positive or negative) γ_k, $k = 1, \ldots, r$. If the basis function centers are chosen to be the experiment design points ($r = n$ and $c^k = \theta^k$), then the approximation provides an interpolating fit. Shin et al. (2002) used a Gaussian basis function, $\exp(-\|\theta - c^k\|^2/2\sigma^2)$. Fitting in the noninterpolating case is by least squares.

Radial basis functions can be used with many kinds of experiment designs. Because of the radial symmetry of the functions, the responses are sensitive to scaling of the design points and the axes. This problem is avoided by scaling variables to $+/-1$ and using the same number of levels for each design variable. In a computational study on deterministic response functions, factorial designs generally provided better fit compared with Latin hypercube designs, except, in some instances, near the center of the design space (Hussain et al., 2002). Radial basis function metamodels are easy to code due to the simple form of (9). Example code is provided by (Watlington, 2005).

2.5 Neural network metamodels

Artificial neural networks (ANNs) can approximate arbitrary smooth functions and can be fitted using noisy response values. ANNs were developed to mimic neural processing, and can be implemented on a digital computer or in parallel using networks of numerical processors, whose inputs and outputs are linked according to specific topologies. For an introduction to neural networks, see Másson and Wang (1990).

ANNs used for function approximation are typically multi-layer feedforward networks. Feedforward layered networks have the flexibility to approximate smooth functions arbitrarily well, provided sufficient nodes and layers. This follows from the work of Kolmogorov (1961) whose results imply that any continuous function f can be reproduced over a compact subset by a three-layer feedforward neural network. While there are some approximation schemes using three layers, most approximations use a two layer network structure, with a single hidden layer and a single output node for models having a univariate dependent variable. The overall metamodel is then a linear combination of linear or nonlinear functions of the argument vector θ. Figure 3 shows the structure for a two-layer feedforward network. The function t is a monotone threshold function, $t(u) \to 0$, $u \to -\infty$ and $t(u) \to 1$, $u \to \infty$. The symbol denotes the element by element weighting of elements of the parameter vector θ by weight coefficients ω. This can be a simple dot product, e.g., $\omega_1 \otimes \theta = \sum_i \omega_{1,i}\theta_i$. The transition value for u is the threshold δ_k.

Commonly used threshold functions include the sigmoid functions: $t(u) = 1/2 + \arctan(u)/\pi$, $t(u) = 1/(1 + \exp(-u))$, $t(u) = 1/2 + \tanh(u)/2$.

While t functions are usually threshold functions, it is useful to imagine more general functions, and to think of neural networks as a technique for computing metamodel coefficients and predicted values for a broad class of metamodels, rather than as representing a particular class of modeling techniques. For example, if the t functions are products of power functions of the θ's, then the model will be a polynomial regression, with λ_k values corresponding to the usual β coefficients.

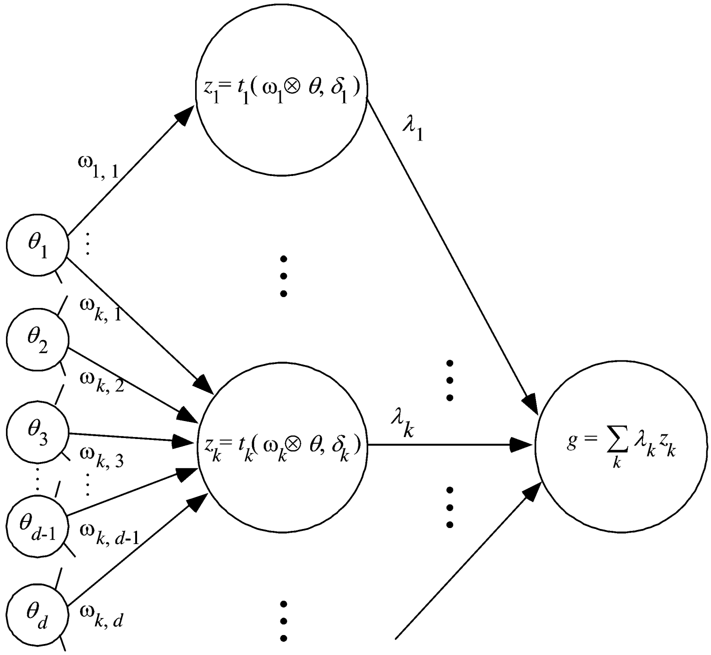


Fig. 3. General structure for two-layer feedforward neural network.

There are many sources for neural network code. See, for example, MATLAB (2005) and Netlab (2005).

2.6 Validating and assessing metamodel fits

Response surface models are validated by a statistical test for lack of fit. This test requires repeated simulation runs under the same parameter setting, e.g., $\theta^i = \theta^{i+1} = \cdots = \theta^{i+r-1}$, where r is the number of replications. This permits the construction of a pure error mean square term, which can be compared with the lack of fit mean square using an F test.

For other metamodel types, fit is usually determined by cross-validation (Stone, 1974). Let $\{y, \theta\}$ represent the experimental results used for fitting, with fitted metamodel function $g(\theta)$ and suppose that $\{y_{-k}, \theta_{-k}\}$ composes the set of experimental results less the kth design point (all replicates) and $g_{-k}(\theta^k)$ is the cross-validation predictive value from the model fitted to $\{y_{-k}, \theta_{-k}\}$. The difference $g_{-k}(\theta^k) - g(\theta)$ can be computed for each design point k, and the sum of squares used as an error measure. Meckesheimer et al. (2002) explored how well the cross-validation error measure approximates the mean squared error of the metamodel fit. The study was restricted to deterministic simulation output using designs with no replications. For radial basis functions the approximation was reasonably good, but for the spatial correlation model, a 'leave-r–out' cross-validation produced a better approximation, where $r = \sqrt{n}$. For additional details on metamodel fitting and validation issues, see Kleijnen and Sargent (2000).

3 Metamodel-based optimization

Law and Kelton (2000) provide a table showing simulation optimization technologies included with commercial simulation software. The list (repeated with web sites in Fu, 2001) shows heuristic search methods including genetic algorithms, tabu search and simulated annealing, but does not include any optimizers solely based on metamodels. Law and McComas (2002) show a similar list in their empirical study. Neural network global metamodels are used in several of these optimizers to screen unpromising new points generated by the heuristic search or to suggest new points to evaluate via simulation (April et al., 2003). For example, see the description of the OptQuest algorithm in Glover et al. (1996).

While metamodels play a role in some of these optimizers, none optimize the metamodel function directly, in a way that might be considered metamodel-based optimization. Further, neither the textbook of Law and Kelton (2000) nor that of Banks et al. (2005) discuss metamodel-based optimization. Fu (2003) states "it is a little baffling that sequential RSM using regression – very well established in the literature *and* quite general and easy to implement – has not been incorporated into any of the commercial packages".

Why then should one be interested in this approach? Metamodel-based optimization simplifies dealing with issues that complicate direct optimization of the simulation model, such as multiple local optima, multiple objectives, and constraints on design parameters and/or responses. This is because the implicitly represented stochastic response of the simulation is replaced by an explicit deterministic metamodel response function. Techniques developed for deterministic optimization can be applied to these metamodel objectives. For example, see Boender and Rinooy Kan (1987), Floudas and Pardalos (1996) and Grossman (1996) for multiple local optima, Charnes and Cooper (1977), Zionts (1992) and Thurston et al. (1994) for handling multiple objectives, and Bazaraa et al. (1993) for handling constraints. Response surface models have substantial statistical theory behind them that permit assessment of the uncertainty about the exact value of the optimal design parameter values and the optimal response (see Myers and Montgomery, 2002; del Castillo and Cahya, 2001; Peterson et al., 2002). Further, the metamodels used during the optimization phase have other usefulness: they can provide insight on the behavior of the simulated system, sensitivity analysis, and the ability to do repeated "what if" analyses quickly. Rapid reporting of the response impacts the efficiency, effectiveness and satisfaction of human interactive design using repeated what if analyses (Simpson et al., 2003).

A significant advantage of a metamodel-based optimization strategy is the incorporation of knowledge of the smoothly varying response function. The metamodel enables a reduction in prediction variance by extending the effect of the law of large numbers over all points in the fitting design. That is, the prediction variance at a design point is less than that which results from an estimate based solely on the replicated simulation runs made at that point. This advantage comes at a cost: bias that is introduced when the metamodel fails to capture the true nature of the response surface (see Figure 2).

A metamodel-based optimization strategy consists of choosing a metamodel form, designing an experiment to fit the metamodel, fitting the metamodel and validating the quality of its fit, optimizing the metamodel (or using it to provide a search direction), and checking the performance of the simulation at the metamodel-predicted optimum (or in the metamodel-determined search direction). In some cases this process is repeated, with the new experiment design focused on the neighborhood of the predicted optimum. Two general strategies have been used for metamodel-based simulation optimization: global metamodel fit, followed by optimization, and iterated local metamodels. These strategies are illustrated graphically in Figure 4. A third strategy has been used with deterministic simulations: global metamodel fit with local updates (Alexandrov et al., 1998).

The iterative local metamodeling strategy is commonly used to determine an optimization search direction. This is followed by a line search using the simulation response directly. Because the metamodels are local, Taylor's theo-

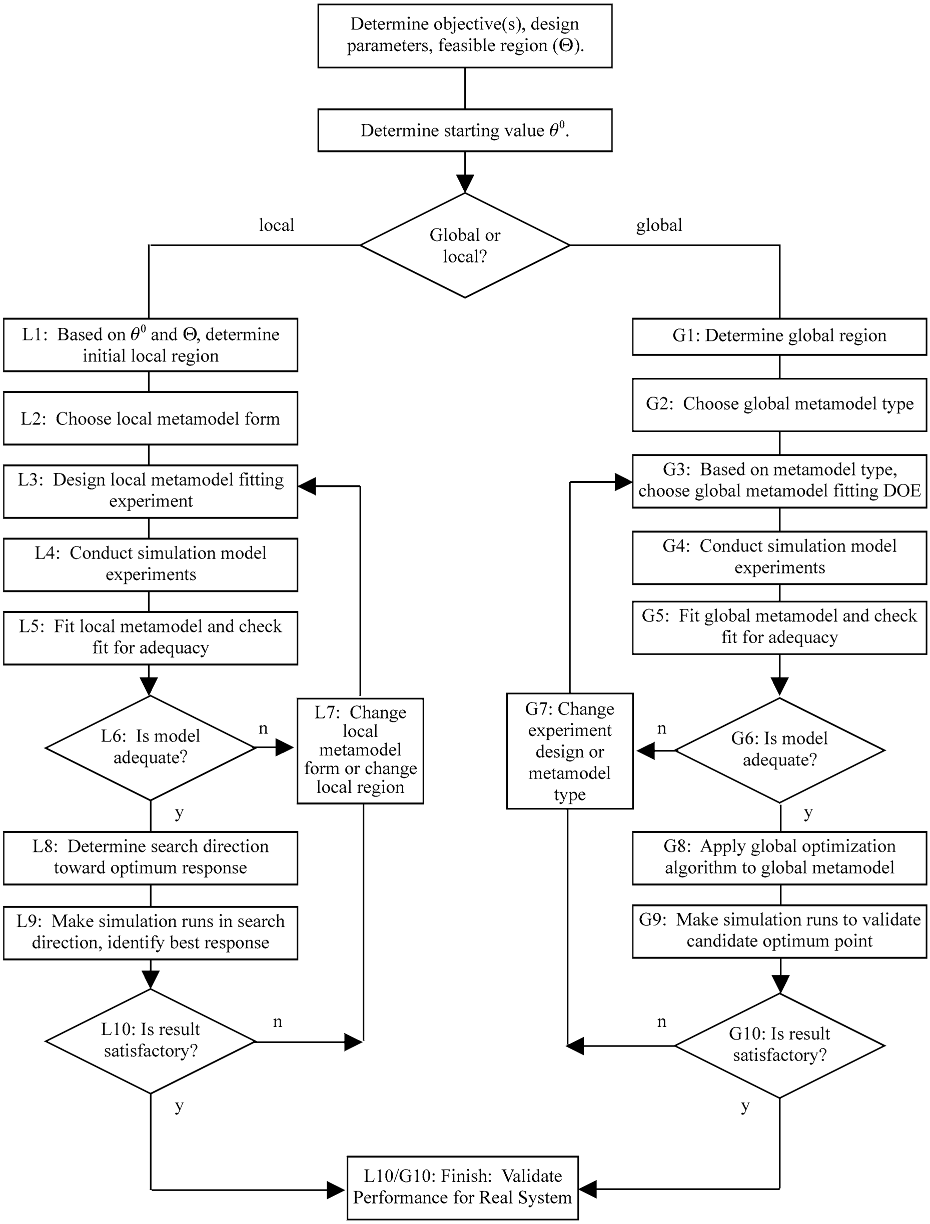

Determine objective(s), design parameters, feasible region (Θ).
Determine starting value θ^0.
Global or local?
local
global
L1: Based on θ^0 and Θ, determine initial local region
L2: Choose local metamodel form
L3: Design local metamodel fitting experiment
L4: Conduct simulation model experiments
L5: Fit local metamodel and check fit for adequacy
L6: Is model adequate?
n
L7: Change local metamodel form or change local region
y
L8: Determine search direction toward optimum response
L9: Make simulation runs in search direction, identify best response
L10: Is result satisfactory?
n
y
G1: Determine global region
G2: Choose global metamodel type
G3: Based on metamodel type, choose global metamodel fitting DOE
G4: Conduct simulation model experiments
G5: Fit global metamodel and check fit for adequacy
G6: Is model adequate?
n
G7: Change experiment design or metamodel type
y
G8: Apply global optimization algorithm to global metamodel
G9: Make simulation runs to validate candidate optimum point
G10: Is result satisfactory?
n
y
L10/G10: Finish: Validate Performance for Real System

Fig. 4. Global and local metamodel-based optimization strategies.

rem implies that linear and quadratic polynomial models can provide adequate fit. This is the scenario for response surface methodology. Of course, determining the meaning of 'local' is critical to the adequacy of these metamodels and to the success of the method. If the local region is too small, response surface characteristics will be swamped by variation in the simulation model output. If the local region is chosen too large, linear or quadratic approximations will be inadequate.

Global metamodel fits using polynomial response surface metamodels are rarely adequate. Instead, spline, neural network, spatial correlation or radial basis function approximation is recommended. The experiment designs for fitting global approximation metamodels differ from the central composite, factorial and fractional factorial designs of RSM optimization. Orthogonal array-based Latin hypercubes tend to perform well. Since global metamodels can have multiple local optima, a global optimization strategy is recommended.

4 Response surface methodology (RSM)

4.1 Origins and strategy of RSM

Response surface methodology has its origins in the work of Box and Wilson (1951). Their collaboration initiated at a chemical company when solving the problem of determining optimal operating conditions for chemical processes. Response surface methodology is used in many practical applications in which the goal is to identify the levels of p design factors or variables, $\theta = (\theta_1, \theta_2, \ldots, \theta_p)$, that optimize a response, $f(\theta)$, over an experimental region. One of the earliest applications in simulation was by Biles (1974). Other early papers on RSM in a simulation context are referenced by Kleijnen (1975b). Since in simulation, RSM uses linear and quadratic model approximations to the simulation model, it is a metamodel-based optimization method. In simulation applications, the system response is obtained from simulation output data. We represent the simulation model outputs by the vector-valued function, as in (2).

In the previous section we introduced a general metamodel-based optimization strategy, which is similar to the formal RSM algorithms described in Neddermeijer et al. (2000) and Nicolai et al. (2004). Table 2 shows how the general strategy is applied specifically for RSM. The labels (L1–L10) map back to the general local metamodel based optimization strategy, illustrated graphically in Figure 4.

The strategy in RSM is to sequentially explore small (local) subregions of the experimental region and use line searches to find a new experimental subregion closer to the optimum. In this approach, first or second-order polynomial models are fit to observed values, y, of the system response. The observed system response values are obtained by means of an experiment, designed to provide a good model fit. The choice of models and designs is such that a series of first-order polynomial models are utilized initially, in order to approach a region in design space that is close to an optimum. This sequence of local approximations using first-order metamodels followed by line searches

Table 2.
RSM strategy for metamodel-based optimization

Phase I: First-order regression	Phase II: Second-order regression
L1: Determine initial local region	
Small enough so linear approximation adequate, large enough so expected effects will be significant	NA
L2: Choose a local metamodel form. See Section 4.2	
First-order polynomial	Second-order polynomial
L3: Design local metamodel fitting experiment. See Section 4.3	
Fractional factorial plus center point	Central composite, small composite or augmented fractional factorials
L5/L6/L7: Fit local metamodel and check fit for adequacy. Change model if necessary. Lack of fit test and tests for significance of regression coefficients. See Section 4.4	
See Figure 5(a)	See Figure 5(b)
L8/L9: Provide a search direction or optimize the metamodel. See Section 4.5	
Steepest ascent/descent	Direction based on canonical/ridge analysis
L10: Check the performance of the simulation at the metamodel-predicted optimum. Confirmation runs. See Section 4.6	

is often called Phase I (see left column in Table 2). When close to the optimum, one or more iterations using second-order models are used to optimize the response function (Phase II of the RSM approach). New experimental designs are constructed or augmented at the line search optimal point. In addition, the use of replications at a center point allows a 'pure error' calculation, which permits a check for lack of fit. The different actions to take based on the outcome of the regression analysis are summarized in Figure 5(a) for the first-order regression and in Figure 5(b) for the second-order regression. Each case is discussed in more detail in Section 4.4. The process comes to an end once the iteration-to-iteration improvement is not practically significant and a number of confirmation runs have been conducted to validate the results. More details on each of the steps are given in the sections indicated in Table 2.

4.2 Choosing a local metamodel form (L2)

The multiple regression model for Phase I is a first-order polynomial model,

$$Y(\theta) = \beta_0 + \sum_{j=1}^{p} \beta_j \theta_j + \varepsilon, \qquad \varepsilon \sim \text{i.i.d.}\, \mathrm{N}(0, \sigma^2). \tag{10}$$

	β = 0	β ≠ 0
LOF	AUGMENT DESIGN AND FIT QUADRATIC MODEL	AUGMENT DESIGN AND FIT QUADRATIC MODEL **GO TO PHASE II**
NO LOF	CHOOSE LARGER RANGE FOR A NEW FIRST-ORDER DESIGN -----OR----- INCREASE THE NUMBER OF REPLICATIONS	LINE SEARCH IN NEGATIVE GRADIENT DIRECTION **GO TO L8/L9**

(a)

	β = 0	β ≠ 0
LOF	UNLIKELY - IF THIS OCCURS, CHOOSE SMALLER RANGE FOR A NEW SECOND-ORDER DESIGN	CHOOSE SMALLER RANGE FOR A NEW SECOND-ORDER DESIGN
NO LOF	UNLIKELY - IF THIS OCCURS, INCREASE THE NUMBER OF REPLICATIONS	LINE SEARCH IN DIRECTION BASED ON CANONICAL ANALYSIS **GO TO L8/L9**

(b)

Fig. 5. Actions based on model adequacy for RSM-based optimization. (a) Model adequacy for Phase I; (b) Model adequacy for Phase II.

Similarly, a full second-order model for Phase II is

$$Y(\theta) = \beta_0 + \sum_{j=1}^{p} \beta_j \theta_j + \sum_{i=1}^{p} \sum_{k=i}^{p} \beta_{ik} \theta_i \theta_k + \varepsilon,$$

$$\varepsilon \sim \text{i.i.d.}\, \mathrm{N}\big(0, \sigma^2\big). \tag{11}$$

The quantitative variables θ are often replaced by coded variables, which are typically scaled to $+/-1$,

$$x_i = \frac{\theta_i - (\theta_{i_{\min}} + (\theta_{i_{\max}} - \theta_{i_{\min}})/2)}{\theta_{i,\max} - (\theta_{i_{\min}} + (\theta_{i_{\max}} - \theta_{i_{\min}})/2)} \quad \text{for } i = 1, \ldots, p.$$

Let D denote the *design matrix*, which is different from the matrix of design parameter values used in the fitting runs $(\,\theta^1 \ldots \theta^i \ldots \theta^n\,)'$. Each column in the matrix corresponds to the function to be multiplied by the corresponding β coefficient in (10) or (11), say $\phi_i(\theta)$ for coefficient β_i. Even for a first-order polynomial regression shown in (10), D and $(\,\theta^1 \ldots \theta^i \ldots \theta^n\,)'$ are not the same; D is augmented with an initial column of ones for the intercept term (i.e., coefficient β_0), as shown in Equation (12). For the $p = 2$ case with n experiment runs,

$$(\,\theta^1\,\ldots\,\theta^i\,\ldots\,\theta^n\,)' = \begin{pmatrix} \theta_1^1 & \theta_2^1 \\ \theta_1^i & \theta_2^i \\ \theta_1^n & \theta_2^n \end{pmatrix},$$

$$D = \begin{pmatrix} \phi_0(\theta^1) & \phi_1(\theta^1) & \phi_2(\theta^1) \\ \phi_0(\theta^i) & \phi_1(\theta^i) & \phi_2(\theta^i) \\ \phi_0(\theta^n) & \phi_1(\theta^n) & \phi_2(\theta^n) \end{pmatrix} = \begin{pmatrix} 1 & \theta_1^1 & \theta_2^1 \\ 1 & \theta_1^i & \theta_2^i \\ 1 & \theta_1^n & \theta_2^n \end{pmatrix}, \tag{12}$$

where scaled x values could be substituted for θ values throughout.

The multiple regression metamodel that is constructed assuming a true response of the form (10) or (11) substitutes 0 for ε and estimates (denoted by b's) for the unknown β coefficients. The b vector is calculated using an existing set of (x, y) data, where x_j^i is the value of the jth design parameter $(j = 1, 2, \ldots, p)$ in the ith run of the system $(i = 1, 2, \ldots, n)$. Let x^i denote the vector of values for the ith run. Finally, y^i is the value of the response in the ith run of the system.

For the single response case, under the assumption of independent, identically distributed random perturbations from run to run, b is found by solving the least-squares equations,

$$b = (D'D)^{-1}D'y. \tag{13}$$

For many simulation situations, the usual assumption $\varepsilon \sim \text{i.i.d.}\,\mathrm{N}(0, \sigma^2)$ does not hold. In many cases this is because the variance increases with the mean. In some cases it is by deliberate intent, through the use of common and antithetic random numbers, for example. In this more general setting normality is still assumed, but the perturbations can be dependent and have different variances. In this case one has $\varepsilon \sim \mathrm{N}(0, \Sigma_Y)$, where Σ_Y is the variance–covariance matrix for the ε values. The vector β can then be estimated using weighted least squares with weight matrix $W = \Sigma_Y^{-1}$

$$b = (D'WD)^{-1}D'Wy.$$

In most cases Σ_Y is unknown and W is an estimate of Σ_Y^{-1} based on sample data.

In some cases, a transformation of the response produces i.i.d. error. There are a number of transformations that can be used for variance stabilizing purposes and to improve the analysis. A family of power transformations has been proposed by Box and Cox (1964). These transformations are of the form

$$y^\lambda = \begin{cases} \frac{y^\lambda - 1}{\lambda}, & \lambda \neq 0, \\ \log(y), & \lambda = 0, \end{cases}$$

where λ typically takes on values of -1 (reciprocal), 0 (log), 1/2 (square root) and 2 (square). Statistical software can estimate the value of λ for the Box–Cox transformation by the method of maximum likelihood. Alternatively, a variance stabilizing transformation can be selected via a plot of log standard deviation of response vs. log mean over all design points. See Chapter 3 of Montgomery (2001) for a detailed discussion of variance-stabilizing transformations. In practice, the analysis is conducted with the transformed response. The results are then transformed back to their original scale for easier interpretation. Transformations of the response are also discussed in Kleijnen (1987).

4.3 Designing local metamodel fitting experiments (L3)

RSM estimates first and second derivatives by regression over a finite region. The concept is similar to the idea of local vs. infinitesimal sensitivity analysis. RSM uses a local measure by running the simulation at a number of points, often with repeated runs at each point (replications) because the response is stochastic. This set of points is the experiment design for that particular RSM step. There are a number of issues to be addressed in order to create such a fitting data set. For simulation experiments, these include (i) choosing the experimental conditions (the θ^i's), (ii) choosing the simulation run lengths, and (iii) choosing the pseudo-random number assignment strategy within and across runs.

There are many criteria for designing a fitting experiment. In the context of response surface methodology:

- the design should result in a good fit of the model to the data in a minimum number of experimental runs,
- it should control errors due to both variance and bias,
- it should allow models of increasing order to be constructed sequentially and allow experiments to be conducted in blocks to accommodate Phase I and Phase II models, and
- it should be robust to the presence of outliers in the data, should allow for lack of fit testing and should provide an estimate of experimental error.

Equations (10), (11) and (13) imply that b can be characterized as a random variable with $\mathrm{E}(b) = \beta$, with variance–covariance matrix

$$\Sigma_b = \sigma^2 \bigl(D'D\bigr)^{-1}$$

and variance of a predicted value at θ_0, say based on b,

$$\mathrm{Var}\bigl(g(\theta_0)\bigr) = \sigma^2 \phi(\theta_0)' \bigl(D'D\bigr)^{-1} \phi(\theta_0),$$

where the ϕ terms are the functions to be multiplied by the corresponding β coefficient as in (12).

Many of the measures of experiment design goodness, that is, the goodness of $(\,\theta^1 \ldots \theta^i \ldots \theta^n\,)'$, attempt to minimize some measure associated with Σ_b or $\mathrm{Var}(g(\theta_0))$. For example, a confidence ellipsoid for the true vector β has a form based on Σ_b

$$(\beta - b)' \bigl(D'D\bigr)^{-1} (\beta - b) = K_\alpha,$$

where the constant K_α depends on the confidence level desired, $100(1-\alpha)\%$. Minimizing the volume of this ellipsoid corresponds to maximizing the determinant of $(D'D)$. This is one measure of design goodness. For the non-i.i.d. case one can substitute $(D'WD)$ for $(D'D)$ in the equations above.

Phase I Designs. Common first-order RSM designs include factorial designs, which comprise full and fractional factorial designs. In factorial designs, each dimension of the design space is covered by a series of (typically) uniformly spaced values; their Cartesian product provides a map of the entire response surface of the system. Usually, first-order designs have only two levels for each design parameter. The number of points required in full factorial designs becomes prohibitively large as the number of factors in the model increases, so fractional factorials are often used as an efficient alternative to full factorial designs when there are many design parameters.

Phase II Designs. Common second-order designs include the central composite (CCD, Box and Wilson, 1951), Box–Behnken (Box and Behnken, 1960), small composite (Draper and Guttman, 1986; Draper and Lin, 1990) and three-level experimental designs (Morris, 2000). A CCD is a two level 2^{p-k} or 2^p factorial design, augmented by n_0 center points and two axial points, so that the quadratic effects can be estimated. This design consists of $2^{p-k} + 2p + n_0$ total design points to estimate $2p + p(p - 1)/2 + 1$ model coefficients. Central composite designs have been used effectively for polynomial models with interaction terms; however, they become impractical for a large number of factors, since they are based on expensive factorial designs. Myers and Montgomery (2002) provide a thorough discussion on response surface modeling and application of factorial designs and other geometric design strategies. The Box–Behnken designs are combinations of incomplete block designs that require fewer levels than CCDs. Small composite designs combine axial runs with fractional-factorials. Morris designs are specially constructed fractional-factorial designs. The designs presented by Morris (2000) are easy to construct and have excellent properties.

Note that there is a significant difference in the number of runs required to fit a ‘first-order’ (linear approximation) vs. the number of runs to fit a quadratic approximation. In terms of the minimum number of runs required, each requires as many runs as there are terms in the model. For a first-order model, there are $p + 1$ terms (including the intercept). For a quadratic model, there are a total of $(p+1)(p+2)/2$ terms. For an optimization on seven factors, the linear approximation requires 8 runs and the quadratic model requires 36 runs, not counting replications. For this reason, an important part of the RSM strategy is to use a linear approximation whenever it is adequate. In particular, when the optimization begins, there is no reason to expect that the initial design parameters are near optimal. If the initial point is far from optimal, the gradient direction may provide an adequate search direction, and a linear approximation may provide an adequate fit.

Replicate runs and variance reduction methods

Another important aspect of experimental design is the use of replicates. One consideration when using replicates is determining at which points those replicates should be conducted. While replicates are often done at the center point of a factorial design to provide an estimate of error and detect curvature, there may be situations for which it may be more appropriate to add replicates at some of the factorial points. For example, in the presence of nonhomogeneous variance in the design space, Kleijnen (2005) suggests increasing the number of replicates to reduce the magnitudes of the variances to reduce the noise at single replicates. In earlier work, Kleijnen and van Groenendaal (1995) propose selecting the additional replicates such that the variances of average responses become a constant. Alternatively, one might consider adjusting the run length of the simulation and partition that run into subruns to obtain replicates (Law and Kelton, 2000). These strategies to reduce variance at high-variance design points are plagued by the $\sqrt{n}$ rule: standard deviations decrease only in the square root of the number of replications (or total simulation run length). If the simulation modeler can tolerate higher prediction error in regions with high response means, then variance stabilizing transformations provide a less costly way to achieve equal variance across the design region.

Variance reduction techniques for simulation models are discussed by Donohue (1995) and McAllister et al. (2001), for example. However, some of these strategies may affect the optimality properties of the experimental design, and the final choice of design strategy (including the number and location of replicates) will depend on the computational expense of the simulation model, the objective and the assumptions of the analysis. Myers and Montgomery (2002) provide a more extensive discussion on the effect of replicates and the design choice.

Control of the random numbers in a simulation permits additional manipulation of the variance/covariance structure of the responses. If RSM is being applied to a discrete-event stochastic simulation model, there are special strategies for selecting the random number streams to improve the precision and accuracy of the fitted model. The original paper in this area is Schruben and Margolin (1978). Donohue et al. (1993a, 1993b, 1995) also discuss designs for fitting quadratic models to discrete-event simulation data. The statistical aspect of analysis with induced correlation and/or control variates are described in Nozari et al. (1987) and Tew and Wilson (1992, 1994). See also Donohue (1995).

4.4 Assessing the adequacy of the metamodel fit (L5/L6/L7)

Testing the model adequacy for RSM is done using a lack-of-fit test, a statistical test that compares model fitting error with “pure” variation within replicated observations. This measures the adequacy of the response surface model. The validation strategy is illustrated in Table 2 and Figure 5, and depends on whether the RSM is in Phase I or Phase II.

For Phase I, if there is no lack of fit, and the regression coefficients are statistically significant ($\beta \neq 0$), a new line search begins in the direction suggested by the regression coefficients. If there is lack of fit, or lack of significance of the model, other steps must be taken.

If there is lack of fit, the design should be augmented with additional runs to fit a local quadratic approximation. Either a central composite design or a small composite design is usually used. An alternative is to use a Resolution V 2^{p-k} design combined with $2p$ axial points and a center point. Such designs are called small composite designs (Draper and Guttman, 1986). Optimization steps using quadratic metamodels are considered to be Phase II of RSM.

On the other hand, if there is no lack of fit, but there is no statistical significance for the first-order regression model, then additional replications should be taken at the fractional factorial points, or a new first-order design should be constructed with a larger range for the design variables. The possible actions following the analysis of the first-order regression model are summarized in Figure 5(a).

For Phase II, cases where there is lack of fit or lack of model significance require a modification of the design: either more replications, or a smaller range for the design.

If some or all of the quadratic model coefficients are statistically significant and there is no lack of fit, then the search direction is selected based on the nature of the quadratic model, as described in Section 4.5. The possible actions following the analysis of the second-order regression model are summarized in Figure 5(b).

4.5 Conducting simulation runs in the search direction (L8/L9)

For Phase I, when the analysis shows that the first-order model is adequate, the regression coefficients are used to identify the gradient. For a minimization problem, the line search proceeds in the negative gradient direction (path of steepest descent), typically beginning at the center of the fractional factorial design region. The path of steepest descent is the direction perpendicular to the contour lines of the response surface. The strategy is to conduct a series of experimental runs along the path of steepest descent indicated by the vector of estimated response surface coefficients b. The step size is usually chosen as the distance from the center of the design region to the edge of the design region. Additional steps are taken in this direction until no further reduction in the objective function occurs. Since the response is stochastic, an unusually good or unusually bad observed value might lead to a premature termination of the search. To reduce this chance, Myers and Khuri (1979) recommend a hypothesis test for significance of the change. del Castillo (1997) and Miró-Quesada and del Castillo (2004) fit a (univariate) polynomial to the responses along the search direction and use the predicted minimum as the end of the line search. They also show that the stopping rule based on three consecutive observations without a decrease is effective.

If multiple first-order fits and line searches are performed, conjugate-gradient or quasi-Newton directions might be chosen instead of the gradient direction (see Joshi et al., 1998).

Phase II generally follows a sequence of one or more Phase I iterations. Eventually the first-order model fails to fit the local response function adequately, and the design is augmented to fit a second-order model for the current design region. This initiates Phase II of the method.

During the analysis of the second-order response surface model, it is sometimes convenient to translate the variables to a new center and rotate the axes so that they correspond to the principal axes of the matrix of second derivatives of the quadratic approximation to the underlying response function, also called the Hessian. The new center is the stationary point of the quadratic model (where the gradient is zero), and the process of rotation and translating the axes to the stationary point is called canonical analysis.

A canonical analysis permits an easy characterization of the local shape of the response function and helps determine whether the estimated stationary point is a maximum, a minimum, or a saddle point. If the eigenvalues of the Hessian are all positive, then the quadratic approximation is bowl-shaped, and a predicted minimum exists, which can be found by setting the gradient of the quadratic approximation equal to zero and solving for x^*. The search direction in this case is toward x^*. If the eigenvalues of H are not all positive, then the search is in the direction of the negative gradient of the fitted quadratic, or may be a ridge direction: a direction that gives the best predicted value of the quadratic on a hypersphere of fixed radius (the radius is typically chosen to be 1 or $\sqrt{n}$ if the factorial values of the variables have been scaled to $+/-1$ over the fitting design).

After a line search is completed, RSM may terminate, either because the budget of runs has been exhausted, or because the optimal point of the most recent line search is very close to the optimal point of the previous cycle.

4.6 Validation of the optimum: checking performance (L9/L10)

Once a local optimum has been found, a number of confirmation runs should be conducted to validate the results. In addition, Peterson et al. (2002) and del Castillo and Cahya (2001) discuss the computation of confidence regions on the stationary point of a response surface. A confidence region provides a measure of quality for the point estimate and can be useful when analyzing problems with multiple responses. Furthermore, confidence regions indicate how robust the solution is. This may be of advantage in determining whether a new metamodel-fitting experiment design, centered about the optimum, should be conducted to refine knowledge of the response function near the optimum.

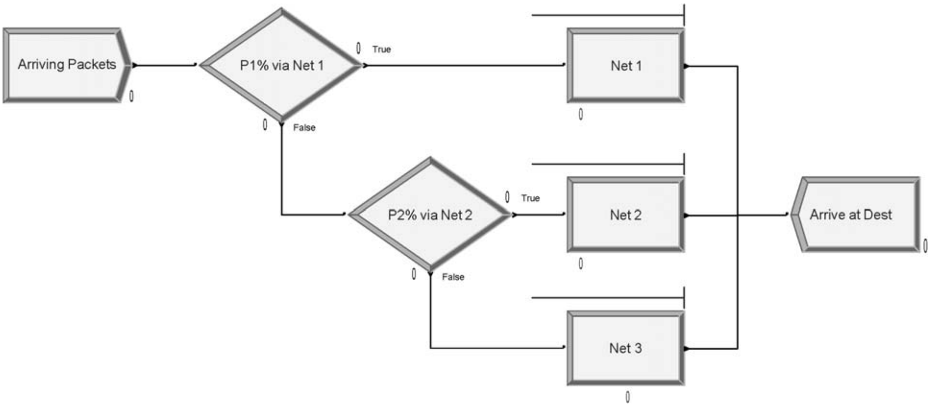


Fig. 6. Arena model for the queuing system design example.

4.7 *Illustration of response surface methodology: queuing example*

There are many operational and strategic issues associated with conducting RSM-based simulation optimization. These are more easily understood in the context of an example, which follows the strategy outlined in Table 2.

Consider a simple network design situation consisting of a communication system in which one must choose routing percentages to route 1000 randomly arriving messages to a particular destination. Suppose that there are three routes (networks) that might be used. One must choose P_1, the percent to network 1 and P_2, the percent of the remaining information packets that go to network 2, to minimize costs. Suppose that costs are composed of \$0.005/time unit each packet is in the system, plus a per-packet processing cost, c_i, that varies by network: \$0.03 for network 1, \$0.01 for network 2 and \$0.005 for network 3. In terms of the general simulation optimization problem, $\theta = (\,P_1\ P_2\,)'$ and $\Theta = [0, 100] \times [0, 100]$, and f is the expected total cost. An Arena model for this system is shown in Figure 6. The Arena simulation environment is described by Kelton et al. (2004). Suppose that packet interarrival times have an exponential distribution with mean $= 1/\lambda = 1$ time unit. Suppose that network transit times have triangular distributions with mean $\mathrm{E}(S)$ and limits $+/-0.5$ with $\mathrm{E}(S) = 1, 2$ and 3 for networks 1, 2 and 3, respectively.

Selecting the design region (L1)

Preliminary simulation runs have revealed that very low as well as very high percentages produce high traffic intensities and high cost. In addition, the initial local metamodel design region should be small enough so a linear approximation is adequate and large enough so that expected effects will be significant. Therefore, the initial region was chosen as $[37, 78] \times [37, 78]$.

Choosing local metamodel type (L2)

The objective of the queuing system analysis is to find values for $\theta = (P_1\ P_2)'$ that minimize the total cost of the system. Suppose that the current settings for P_1 and P_2 are 40% and 75% (the center of the initial design region), and we suspect that the minimum cost is not near the current settings. Therefore, the local surface could be approximated by a first-degree polynomial

$$Y = \beta_0 + \beta_1\theta_1 + \beta_2\theta_2 + \varepsilon, \quad \varepsilon \sim \mathrm{NID}(0, \sigma^2).$$

Designing local metamodel fitting experiment: Phase I (L3)

The design consisted of 10 simulation runs; a 2^2 factorial design with 2 replications at each point was chosen as a first-order design to fit this model. Replicates were used to provide an estimate of the experimental error. In addition, the inclusion of runs at a center point permits a check on the adequacy of the first-order model. The design as well as the data obtained from it are shown in Table 3.

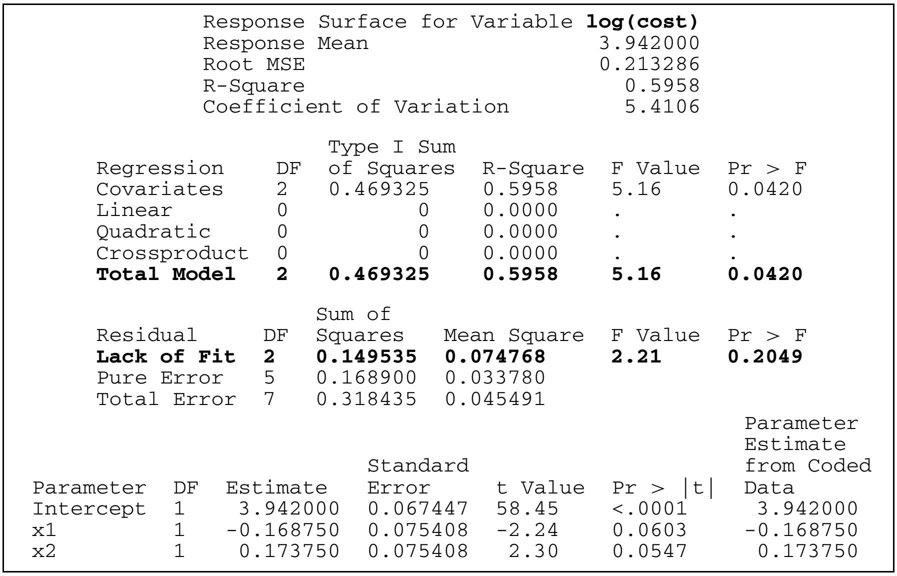
Examining the standard deviations of the response (*cost*) at each of the design points shows some indication of nonhomogeneous standard homogeneous deviations (12.10, 3.26, 14.44, 9.60 and 4.13). A transformation may be appropriate to stabilize the variance. A maximum likelihood estimation of the power transformation parameter suggests a log transformation on the response. The transformed response values have also been included in Table 3. Subsequent analyses for this iteration are done for the transformed response.

Check model adequacy: Phase I (L5/L6/L7)

The analysis of variance and lack of fit for the least squares analysis is summarized in Figure 7. The first-order model is significant, indicated by a P-value of 0.0420 and there is no lack of fit, indicated by the P-value of 0.2049. For this situation, Figure 5(a) suggests continuing with a line search.

Table 3.
Data obtained from a first-order factorial design (centered around $P_1 = 40$, $P_2 = 75$)

Run	Design variables		Coded variables		Response	
	$\theta_1 = P_1$	$\theta_2 = P_2$	x_1	x_2	*cost*	ln(*cost*)
1	37	72	−1	−1	54.40	4.00
2	37	72	−1	−1	37.29	3.62
3	43	72	+1	−1	39.90	3.69
4	43	72	+1	−1	35.29	3.56
5	37	78	−1	+1	64.87	4.17
1	37	78	−1	+1	85.29	4.45
2	43	78	+1	+1	52.83	3.97
3	43	78	+1	+1	39.25	3.67
4	40	75	0	0	66.14	4.19
5	40	75	0	0	60.30	4.10



```
                     Response Surface for Variable log(cost)
                     Response Mean                      3.942000
                     Root MSE                           0.213286
                     R-Square                             0.5958
                     Coefficient of Variation             5.4106

                               Type I Sum
            Regression     DF  of Squares   R-Square   F Value   Pr > F
            Covariates     2   0.469325     0.5958     5.16      0.0420
            Linear         0          0     0.0000     .         .
            Quadratic      0          0     0.0000     .         .
            Crossproduct   0          0     0.0000     .         .
            Total Model    2   0.469325     0.5958     5.16      0.0420

                              Sum of
            Residual      DF  Squares     Mean Square  F Value   Pr > F
            Lack of Fit   2   0.149535   0.074768      2.21      0.2049
            Pure Error    5   0.168900   0.033780
            Total Error   7   0.318435   0.045491
                                                                   Parameter
                                                                   Estimate
                                        Standard                   from Coded
     Parameter  DF   Estimate    Error      t Value   Pr > |t|    Data
     Intercept  1     3.942000   0.067447   58.45     <.0001       3.942000
     x1         1    -0.168750   0.075408   -2.24     0.0603      -0.168750
     x2         1     0.173750   0.075408    2.30     0.0547       0.173750
```

Fig. 7. Phase I analysis of first-order regression model with transformed response.

Conducting simulation runs in the search direction: Phase I (L8/L9)

With the coefficient estimates for the first-order model the equation of the first-order model can be written as

$$\ln\bigl(g(x)\bigr) = 3.942 - 0.16875x_1 + 0.17375x_2.$$

Rounding the coefficients for the gradient gives $(\,-0.17\ 0.17\,)'$ which can be scaled to $(\,-1\ \ 1\,)'$. Because the objective is to minimize the response, we take a series of steepest descent steps, starting at the center of the initial experimental region and moving $b_2/b_1 = 1/(-1) = -1.0$ units in x_2 for every 1 unit in x_1. Table 4 illustrates this process. Note how the response initially decreases, but then increases again in run 15. As recommended in Section 5.4, three additional steps were done to verify that the increase was not due to variability in the process.

A new 2^2 factorial design can be constructed in the vicinity of run 14; the design is shown in Table 5. This time, there is much less variability in the standard deviations of the responses and we proceed with the analysis of variance and lack of fit for the least squares analysis without a response transformation. This analysis is summarized in Figure 8.

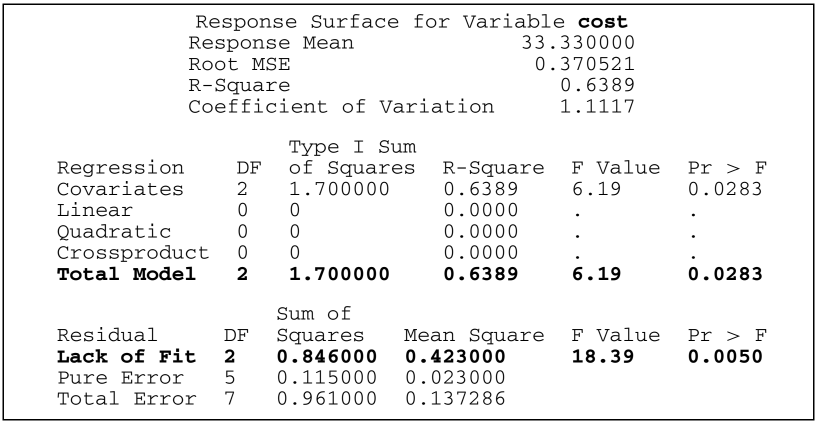
In this case, the model is significant, with a P-value of 0.0283, but the lack of fit test is also significant, with a P-value of 0.005. Following the path suggested in Figure 5(a), the first-order design is augmented to construct a central composite design with axial points at $\delta = \sqrt{2}$, as shown in Table 6.

Table 4.
Steepest descent path for queuing system design

Run	Coded variables		Design variables		Total cost
	x_1	x_2	P_1	P_2	y
Center run conditions	0	0	40	75	63.2 (average)
11	1	−1	43	72	38.3
12	2	−2	46	69	36.2
13	3	−3	49	66	33.3
14	**4**	**−4**	**52**	**63**	**32.6**
15	5	−5	55	60	33.5
16	6	−6	58	57	34.4
17	7	−7	61	54	34.2

Table 5.
Data obtained from a first-order factorial design (centered around $P_1 = 52$, $P_2 = 63$)

Run	Design variables		Coded variables		Total cost
	P_1	P_2	x_1	x_2	y
18	49	60	−1	−1	34.3
19	49	60	−1	−1	34.0
20	55	60	+1	−1	33.5
21	55	60	+1	−1	33.5
22	49	66	−1	+1	33.3
23	49	66	−1	+1	33.5
24	55	66	+1	+1	32.8
25	55	66	+1	+1	32.9
26	52	63	0	0	32.9
27	52	63	0	0	32.6

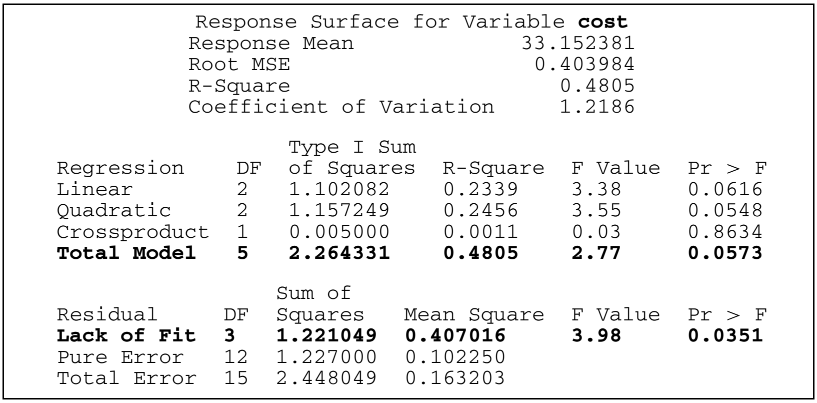


```
                    Response Surface for Variable cost
                   Response Mean                33.330000
                   Root MSE                      0.370521
                   R-Square                        0.6389
                   Coefficient of Variation        1.1117

                        Type I Sum
Regression     DF   of Squares    R-Square   F Value   Pr > F
Covariates      2   1.700000      0.6389     6.19      0.0283
Linear          0   0             0.0000     .         .
Quadratic       0   0             0.0000     .         .
Crossproduct    0   0             0.0000     .         .
Total Model     2   1.700000      0.6389     6.19      0.0283

                   Sum of
Residual       DF  Squares     Mean Square   F Value   Pr > F
Lack of Fit    2   0.846000    0.423000      18.39     0.0050
Pure Error     5   0.115000    0.023000
Total Error    7   0.961000    0.137286
```

Fig. 8. Phase I analysis of first-order regression model, after the line search.

Table 6.
Data obtained from an augmented design (centered around $P_1 = 52$, $P_2 = 63$)

Run	Design variables		Coded variables		Total cost
	P_1	P_2	x_1	x_2	y
28	48	63	−1.41	0	32.7
29	48	63	−1.41	0	33.7
30	56	63	+1.41	0	33.4
31	56	63	+1.41	0	32.8
32	52	59	0	−1.41	32.8
33	52	59	0	−1.41	33.4
34	52	67	0	+1.41	33.2
35	52	67	0	+1.41	32.7
36	52	63	0	0	33.0
37	52	63	0	0	32.7
38	52	63	0	0	32.5



```
                 Response Surface for Variable cost
                Response Mean               33.152381
                Root MSE                     0.403984
                R-Square                       0.4805
                Coefficient of Variation       1.2186

                        Type I Sum
Regression     DF     of Squares   R-Square   F Value   Pr > F
Linear          2     1.102082     0.2339     3.38      0.0616
Quadratic       2     1.157249     0.2456     3.55      0.0548
Crossproduct    1     0.005000     0.0011     0.03      0.8634
Total Model     5     2.264331     0.4805     2.77      0.0573

                     Sum of
Residual       DF    Squares    Mean Square   F Value   Pr > F
Lack of Fit     3    1.221049   0.407016      3.98      0.0351
Pure Error     12    1.227000   0.102250
Total Error    15    2.448049   0.163203
```

Fig. 9. Phase II analysis of second-order regression model (based on CCD).

Check Model Adequacy: Phase II (L5/L6/L7)

The analysis of the second-order response surface model is summarized in Figure 9.

The analysis of the second-order model shows that there is lack of fit, with P-value 0.0351, and marginal model significance, with P-value 0.0573. Referring back to Figure 5(b), this is a case that requires choosing a smaller range for a new second-order design. In an attempt to reduce the cost of exercising the simulation code, we first reduce the range of the second-order design by placing the axial points of the CCD at the center of the faces (with $\delta = 1$), creating a face-centered design. The four new points, with 2 replications each, are shown in Table 7.

The analysis of the second-order model based on the face-centered design is shown in Figure 10.

Table 7.
Data for a face centered design (centered around $P_1 = 52$, $P_2 = 63$)

Run	Design variables		Coded variables		Total cost
	P_1	P_2	x_1	x_2	y
39	49	63	−1	0	33.3
40	49	63	−1	0	33.4
41	55	63	+1	0	32.4
42	55	63	+1	0	32.8
43	52	60	0	−1	33.4
44	52	60	0	−1	33.4
45	52	66	0	+1	33.2
46	52	66	0	+1	32.9

```
                 Response Surface for Variable cost
                Response Mean                33.227778
                Root MSE                      0.168874
                R-Square                        0.9108
                Coefficient of Variation        0.5082

                      Type I Sum
Regression     DF   of Squares   R-Square   F Value   Pr > F
Linear         2    2.288333     0.5965     40.12     <.0001
Quadratic      2    1.200556     0.3130     21.05     0.0001
Crossproduct   1    0.005000     0.0013     0.18      0.6828
Total Model    5    3.493889     0.9108     24.50     <.0001

                    Sum of
Residual       DF   Squares    Mean Square   F Value   Pr > F
Lack of Fit    3    0.097222   0.032407      1.19      0.3672
Pure Error     9    0.245000   0.027222
Total Error    12   0.342222   0.028519
```

Fig. 10. Phase II analysis of second-order regression model.

The adjustment to the design range improved the second-order model with a model significance P-value of less than 0.0001 and a lack of fit P-value of 0.3672. Using Figure 5(b) as a guide, we continue with a line search.

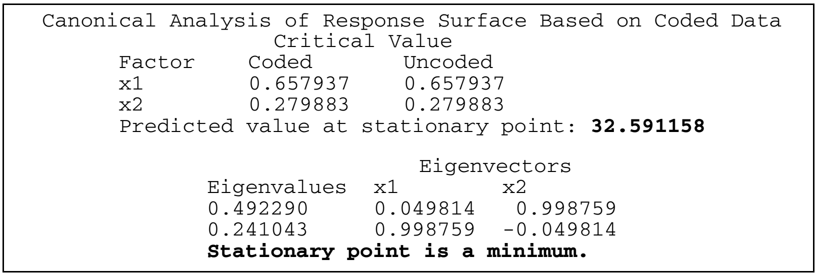
Conducting simulation runs in the search direction: Phase II (L8/L9)

Figure 11 summarizes the results for the canonical analysis, as discussed in Section 4.5.

The stationary point found is a minimum point, with $(\,\theta_1\ \ \theta_2\,) = (\,54.0\ 63.8\,)$, after translating the coded variables $(\,x_1\ x_2\,) = (\,0.66\ 0.28\,)$ to the original units. The estimated cost at the optimal operating conditions was \$32.59.

Conducting simulation runs in the search direction: Phase II (L8/L9)

To confirm the results from the analysis, 16 additional runs were made at $P_1 = 54.0$ and $P_2 = 63.8$. In practice, the number of confirming runs may depend on how many runs one can afford. The 95% confidence interval for the cost based on 16 replications was [\$32.77, \$33.31]. Compared with the



```
Canonical Analysis of Response Surface Based on Coded Data
                  Critical Value
      Factor     Coded        Uncoded
      x1         0.657937     0.657937
      x2         0.279883     0.279883
      Predicted value at stationary point: 32.591158

                             Eigenvectors
             Eigenvalues  x1         x2
             0.492290     0.049814   0.998759
             0.241043     0.998759  -0.049814
             Stationary point is a minimum.
```

Fig. 11. Phase II canonical analysis for the second-order regression model.

average cost at the starting point, \$54.97, the response has improved significantly. We performed a 1000-replication validation for the same point, not likely to be practical in many settings, which provided a confidence interval of [\$33.03, \$33.10] for the cost.

4.8 RSM for simulation optimization

RSM has been used successfully for over 40 years for processes with stochastic variation. It has been successfully applied to stochastic simulation problems for approximately half that time. The advantage of the method is that it is robust. The disadvantages of the method are that automated versions of the algorithm are not readily available, and manual implementation of the method for more than a few cycles is tedious, complex, and prone to error. In Section 5, we use the network design example to illustrate a global metamodel approach and also provide an analytic solution to this problem.

5 Global metamodel-based optimization

5.1 Motivation and strategy

The developments in global approximation model technology present an opportunity for optimization using a single metamodel, rather than a sequence of fitted local metamodels. There are several potential advantages to this approach. First, a relatively flexible global metamodel may be able to provide a high-fidelity approximation for the response surface with relatively few experimental points, while a polynomial (RSM) metamodel using the same experimental data would fail.

Second, the overall process is simplified: there can be a single experiment design, and a single model-fitting step. This removes the need for sequential decisions on the type of metamodel to be fit and the kind of experiment design to be used for fitting.

Of course, more complex global metamodel-based optimization methods could be designed, for example, to update the fit by selecting additional

Table 8.
General and specific global strategy for metamodel-based optimization

General global strategy	Network design example strategy
G1: Determine global region	Smaller than the feasible region: elimination of obvious nonoptimal regions
G2: Choose a global metamodel type	Smoothing spline
G3: Choose global metamodel fitting DOE	Full factorial (4^2) plus center for smoothing spline
G5/G6/G7: Fit global metamodel and check fit for adequacy. Change model if necessary	Leave-one-out cross-validation sum of squared error. Confirmation runs
G8: Apply global optimization algorithm	Grid search
G10: Check the performance of the simulation at the metamodel-predicted optimum	Confirmation runs

simulation runs as the optimization progressed (Alexandrov et al., 1998; Booker et al., 1999). Such strategies are actually a mix of global and local strategies and will not be considered in this chapter. We will assume that the strategy follows that shown in Figure 4, and that the metamodel-fitting step occurs once, rather than having a sequential update of the design and refitting of the model.

In this section we illustrate a global metamodel-based optimization strategy for the network routing example described in the previous section. Table 8 shows the general global strategy from Figure 2 and the specific implementation that will be used for the network routing optimization example. The process is described in detail in the following sections.

5.2 *Selecting the design region (G1)*

Recall that for the network design example the design variables $\theta = (P_1 \quad P_2)$. The objective is to find routing percentages that minimize total cost (network use plus transit time) for 1000 messages. The feasible region is $\Theta = [0, 100] \times [0, 100]$, since the routing percentages can be set to any value. Setting a percentage near 100 means that the subsequent network route(s) will not be used, and so the traffic intensity might be too high on the used network(s). On the other hand, a percentage near zero again means that a network will not be used, and that traffic intensities on the remaining networks may be excessive. Some exploratory simulation runs showed that percentages less than 40 or greater than 80 tended to produce high traffic intensities and high costs. For this reason, the global metamodel fitting region was reduced from $[0, 100] \times [0, 100]$ to $[40, 80] \times [40, 80]$.

5.3 Global metamodel type (G2)

Section 2 describes many possible types for the global metamodel. We selected smoothing splines for several reasons. First, they are widely used for response functions of one or two variables. Second, there is publicly available code for fitting and prediction using smoothing splines (Dierckx, 1981, 1993; NETLIB, 2005). Third, they allow for a weighted fit based on observed standard deviations of responses.

5.4 Experiment design (G3)

Bivariate smoothing splines can be used with a variety of design types. We believed the response would be well-behaved (but not quadratic), with substantial increases in cost near the boundaries of the design region. Beyond that, we had no special knowledge of the likely location of the optimum, and chose a 4^2 factorial design to cover the design space, plus a center point ($P_1 = 60$, $P_2 = 60$).

The design consisted of 34 simulation runs, two runs at each of the 17 design points. Table 9 shows the average cost and standard deviation of cost at each design point. Note that there were significant variations in the observed standard deviations. This problem is more likely to occur when fitting global metamodels than when fitting local RSM-type metamodels.

Table 9.
Means and standard deviations for the 17 point, 34 run 4^2 factorial design

Design point	P_1	P_2	Avg. cost	S.D. cost
1	40	40	92.527	22.335
2	40	53	41.200	5.547
3	40	67	34.350	0.776
4	40	80	109.445	40.906
5	53	40	40.485	3.222
6	53	53	34.413	1.572
7	53	67	33.418	0.242
8	53	80	35.275	1.089
9	67	40	35.483	0.327
10	67	53	35.063	0.715
11	67	67	34.350	0.776
12	67	80	34.661	1.250
13	80	40	43.248	3.612
14	80	53	40.711	0.308
15	80	67	39.953	0.676
16	80	80	39.967	1.320
17	60	60	33.293	0.379

5.5 Checking model adequacy (G5, G6, G7)

We expected that a poor model fit might occur since there was substantial variation in standard deviations across the design points. This is a common characteristic for queueing simulations: means and standard deviations of time in system are typically related. We would like to limit the influence of high-variance observations on the overall fit by considering a variance-stabilizing transformation for the response.

Figure 12 shows the log standard deviation vs. the log mean for the 17 design points, which has a slope of approximately 3. This suggests a variance stabilizing transformation of $1/(cost^2)$, as discussed in Montgomery (2001). Figure 13 shows the fitted global metamodel for the untransformed data, using a weight function equivalent to the inverse standard deviation as well as metamodel fitted to the transformed data and using a constant standard devia-

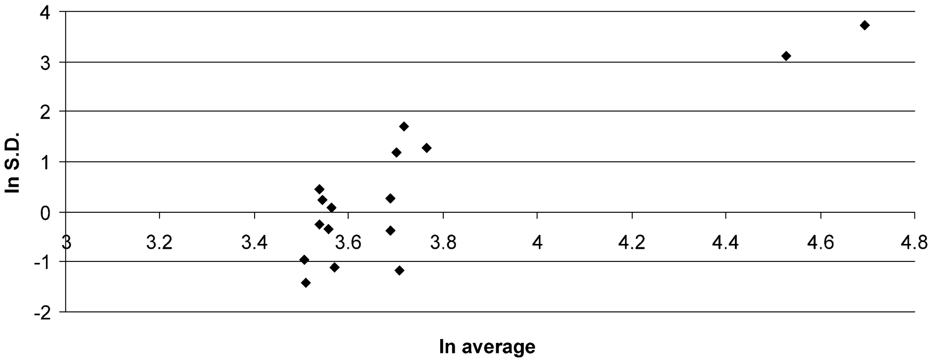
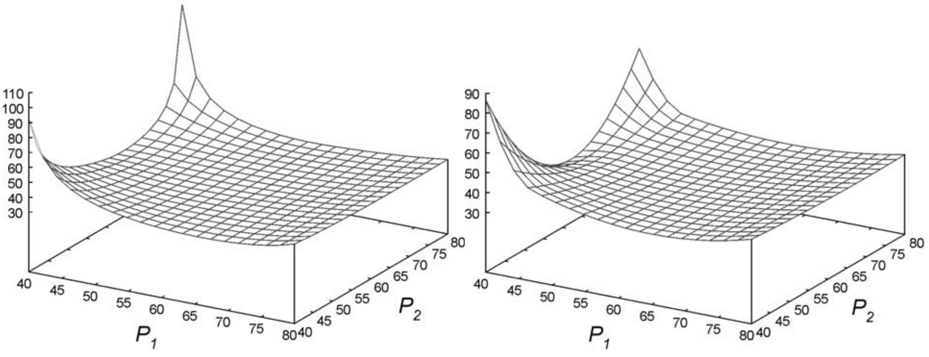


Fig. 12. Log standard deviation vs. log mean for the 17-point 4^2 design.



(a) (b)

Fig. 13. Fitted global metamodels for the 17-point 4^2 design: (a) $1/y^2$ transformation, (b) no transformation.

tion assumption. The order of magnitude of the smoothing parameter in each case was chosen to provide good leave-one-out cross-validation results. For both the transformed (T-Model) and untransformed (U-Model) response, the smoothing parameter was set to 1.7. Although leave-one-out cross-validation has been shown useful in metamodel assessment (Meckesheimer et al., 2002), it has some drawbacks (Shao, 1993; Tibshirani, 1996). For that reason the two model fits were checked against 10 replications of four randomly selected confirmation design points.

The cross-validation and confirmation run results are summarized in Table 10. While the untransformed metamodel provides better cross-validation results, the transformed model provides lower error (though not statistically significant) for the confirmation runs. More than three times as many confirmation runs would be required to detect the observed differences as statistically significant. There is not a strong reason to select one model over the other, but we continue with the transformed response metamodel to parallel the analysis in the RSM section.

Table 10.
Validation results for metamodels with transformed (T-model) and untransformed (U-model) responses

P_1	P_2	Cost	U-Model	U-Error^2	T-Model	T-Error^2
Cross-validation results						
40	67	34.35	37.46	9.69	41.18	46.72
40	80	109.45	46.55	3956.12	35.10	5527.71
40	53	41.20	39.87	1.77	46.34	26.41
40	40	92.53	85.31	52.03	10.70	6695.82
80	67	39.95	39.62	0.11	41.30	1.82
80	80	39.97	41.43	2.15	34.78	26.90
80	53	40.71	41.25	0.29	48.88	66.77
80	40	43.25	42.18	1.13	35.35	62.36
53	67	33.42	32.43	0.98	32.21	1.46
53	80	35.28	39.74	19.98	40.21	24.34
53	53	34.41	35.15	0.54	34.11	0.09
53	40	40.48	38.71	3.15	43.36	8.27
67	67	34.35	34.40	0.00	34.09	0.07
67	80	34.66	33.35	1.71	36.71	4.19
67	53	35.06	34.61	0.20	34.02	1.08
67	40	35.48	36.38	0.81	37.56	4.33
60	60	33.29	33.59	0.09	33.62	0.11
Average cross-validation squared error				238.28		735.20
Confirming run validation results						
53	72	33.89	33.80	0.008	34.18	0.084
40	80	34.71	36.33	2.624	35.82	1.232
40	53	34.18	34.30	0.014	34.57	0.152
40	40	36.02	35.16	0.740	35.47	0.302
Average confirming run squared error				0.85		0.44

5.6 Global optimization results (G8)

The global optimization might proceed with a multistart gradient-based optimizer, identifying the global optimum as in Boender and Rinooy Kan (1987). For a two-variable optimization, a simpler grid search strategy is possible. The optimal operating conditions based on this search are $P_1 = 54.1$ and $P_2 = 64$, with an estimated cost of \$33.34 (for the untransformed response metamodel the results were $P_1 = 57.1$ and $P_2 = 65.6$, with an estimated cost of \$33.14).

5.7 Validation: confirming runs (G9)

Confirming runs of the simulation were made at $P_1 = 54.1$ and $P_2 = 64$. In practical situations it may be cost-prohibitive to conduct many replications. For this example we have chosen a middle ground for replications at 16. This provides a 95% confidence interval of [\$32.75, \$33.22] for the cost. We performed a 1000-replication validation for the same point, not likely to be practical in many settings, which provided a confidence interval of [\$33.05, \$33.11] for the cost. For comparison, the untransformed candidate for optimum produced a 1000-replication confidence interval of [\$33.18, \$33.24], inferior to the transformed response metamodel solution. With 16 replications, the two candidate solutions were indistinguishable.

5.8 An alternative global 'metamodel' based on steady-state behavior

An alternative 'model of the simulation model' is to take the steady-state approximation, which can be solved analytically. This is equivalent to assuming that the network queues begin in steady state, rather than empty and idle.

Suppose that the system to be studied terminates after processing 1000 jobs. The total cost can be approximated using the analytical solution for the steady-state problem. It is possible to decompose the system into three $M/G/1$ queues. In steady state, the average transit time on the ith network will be w_i. If p_i is the probability that a packet is routed through network i and w_i is its average transit time, then the total cost in the steady-state approximation will be

$$\text{Expected total cost for 1000 steady-state customers} \\ = \sum_{i=1}^{3} 1000 p_i (c_i + w_i). \tag{14}$$

The routing probabilities are computed as $p_1 = P_1/100$, $p_2 = (P_2/100)(1 - P_1/100)$ and $p_3 = 1 - [P_1/100 + (P_2/100)(1 - P_1/100)]$. For an $M/G/1$ system in steady state, the average time in the system for an entity is

$$w = \mathrm{E}(S) + \lambda \frac{(\mathrm{E}(S))^2 + \sigma^2}{2(1 - \lambda \mathrm{E}(S))}.$$

The queue for machine 1 has $\lambda_1 = \lambda(P_1/100) = P_1/100$, $\mu_1 = 1/\mathrm{E}(S_1) = 1$ and $\sigma_1^2 = [0.5^2 + 1^2 + 1.5^2 - (0.5)(1) - (0.5)(1.5) - (1)(1.5)]/18 = 1/24$, which gives

$$w_1 = 1 + \frac{(P_1/100)(1 + (1/24))}{2(1 - (P_1/100))}.$$

Similarly,

$$w_2 = 2 + \frac{(P_2/100)(1 - (P_1/100))(4 + (1/24))}{2(1 - ((P_2/100)(1 - (P_1/100))2))}$$

and

$$w_3 = 3 + \frac{(1 - ((P_1/100) + (P_2/100)(1 - P_1/100)))(9 + (1/24))}{2(1 - (1 - ((P_1/100) + (P_2/100)(1 - P_1/100))3))}.$$

By solving these equations for the values of P_1 and P_2 that minimize the expected total cost in Equation (14), we find that the optimum is at \$33.10 with $P_1 = 53.5$ and $P_2 = 63.1$.

The validation runs for this design point give a 1000-replication average of \$33.04 and a 95% confidence interval for the mean of [\$33.01, \$33.07]. One would expect the true optimal routing values to transfer a bit more traffic to routes 2 and 3 than the optimal steady-state solution, with slightly lower cost, since at the start the routes are not congested. This implies that $(\,53.5 \quad 63.1\,)$ may overestimate the percentages. The differences in performance are very small in this neighborhood, however, and not much more can be gained over the design identified by the analytical metamodel. For example the slight reduction of $P_1 = 53.1$ and $P_2 = 62.7$ drops less than \$0.02 in observed average cost for a 1000-replication validation, statistically indistinguishable from the expected cost for the simulation at the analytic solution.

6 Summary

Optimization of computationally costly simulations can be approximated by optimizing metamodels as surrogates for the costly simulation response functions. Local metamodels can be used within an iterative optimization strategy, developed or updated as the optimization progresses. Alternatively, global metamodels can be fitted once, based on a set of simulation runs from a global experiment design, and then the optimization can proceed iteratively using the same metamodel. In either case, it is important to (i) choose the metamodel form carefully, (ii) choose an experiment design appropriate for fitting that type of metamodel, and (iii) validate the metamodel fit and the predicted optimal operating conditions.

Metamodel-based optimization has two distinct benefits over other simulation optimization approaches: a reduction in prediction error that comes from an aggregation of error across many design points, and a representation that often permits insight into the behavior of the response function.

Metamodels can also be used for optimization in a robust design context. In that setting, the objective is to seek an ideal mean response while minimizing its variance. Ramberg (Ramberg et al., 1991), Sanchez (2000) and others cited by these authors discuss the metamodel approach to robust design.

Nearly thirty years after the coining of the term, metamodel-based optimization continues to be an active area of research. A number of researchers are studying how Bayesian methods can improve the fitting and prediction processes (Cheng and Currie, 2004; Chick, 2004), others examine the sequential design of fitting experiments for Bayesian and other metamodels (Kleijnen and van Beers, 2004; Santner et al., 2000, 2003; van Beers and Kleijnen, 2004). Barton (2005) examines alternatives to metamodel-based optimization when this optimization serves as a proxy for having an inverse function metamodel. Myers (1999) gives a fairly recent summary of research issues for RSM.

Finally, general purpose metamodel functions are not always the best approach: when the simulation models are relatively simple, analytic approximations may be used, and can serve as equally effective metamodels.

Acknowledgements

This work was partially supported by NSF Grant No. DMI-0084918. The authors appreciate the suggestions made by the editors, Barry Nelson and Shane Henderson, and by Averill Law. The authors also acknowledge helpful discussions with Andrew J. Booker and Stephen P. Jones from Applied Statistics, Phantom Works, the Boeing Company.

References

Alexandrov, N., Dennis Jr., J.E., Lewis, R.M., Torczon, V. (1998). A trust region framework for managing the use of approximate models in optimization. *Structural Optimization* 15, 16–23.

Andradóttir, S. (1998). A review of simulation optimization techniques. In: Medeiros, D.J., Watson, E.F., Carson, J.S., Manivannan, M.S. (Eds.), *Proceedings of the 1998 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 151–158.

April, J., Glover, F., Kelly, J.P. (2003). Practical introduction to simulation optimization. In: Chick, S., Sánchez, P.J., Ferrin, D., Morrice, D.J. (Eds.), *Proceedings of the 2003 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 71–78.

Banks, J., Carson II, J.S., Nelson, B.L., Nicol, D.M. (2005). *Discrete-Event System Simulation*, 4th edition. Prentice Hall, Upper Saddle River, NJ.

Barton, R.R. (1992). Metamodels for simulation input–output relations. In: Swain, J.J., Goldsman, D., Crain, R.C., Wilson, J.R. (Eds.), *Proceedings of the 1992 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 289–299.

Barton, R.R. (1998). Simulation metamodels. In: Medeiros, D.J., Watson, E.F., Carson, J.S., Manivannan, M.S. (Eds.), *Proceedings of the 1998 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 167–174.

Barton, R.R. (2005). Issues in development of simultaneous forward-inverse metamodels. In: Kuhl, M.E., Steiger, N.M., Armstrong, F.B., Joines, J.A. (Eds.), *Proceedings of the 2005 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 209–217.

Bazaraa, M.S., Sherali, H.D., Shetty, C.M. (1993). *Nonlinear Programming: Theory and Algorithms*, 2nd edition. Wiley, New York.

Bertsekas, D.P. (1999). *Nonlinear Programming*, 2nd edition. Athena, Nashua, NH.

Biles, W.E. (1974). A gradient regression search procedure for simulation experimentation. In: Highland, H.J., Steinberg, H., Morris, M.F. (Eds.), *Proceedings of the 1974 Winter Simulation Conference*. Association for Computing Machinery, New York, pp. 491–497.

Boender, C.G.E., Rinooy Kan, A.H.G. (1987). Bayesian stopping rules for multistart global optimization methods. *Mathematical Programming* 37, 59–80.

Booker, A.J., Dennis, J.E., Frank, P.D., Serafini, D.B., Torczon, V., Trosset, M.W. (1999). A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization* 17, 1–13.

Box, G.E.P. (1954). The exploration and exploitation of response surfaces. Some general considerations and examples. *Biometrics* 10, 16–60.

Box, G.E.P., Behnken, D.W. (1960). Some new three-level designs for the study of qualitative variables. *Technometrics* 2, 455–475.

Box, G.E.P., Cox, D.R. (1964). An analysis of transformations (with discussion). *Journal of the Royal Statistical Society B* 26, 211–246.

Box, G.E.P., Draper, N.R. (1987). *Empirical Model-Building and Response Surfaces*. Wiley, New York.

Box, G.E.P., Wilson, K.B. (1951). On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society B* 13, 1–45.

Breiman, L. (1991). The *Π* method for estimating multivariate functions from noisy data (with discussion). *Technometrics* 33, 125–160.

Charnes, A., Cooper, W.W. (1977). Goal programming and multiple objective optimization – part I. *European Journal of Operational Research* 1, 39–54.

Cheng, R.C.H., Currie, C.S.M. (2004). Optimization by simulation metamodelling methods. In: Ingalls, R.G., Rossetti, M.D., Smith, J.S., Peters, B.A. (Eds.), *Proceedings of the 2004 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 485–490.

Chick, S.E. (2004). Bayesian methods for discrete event simulation. In: Ingalls, R.G., Rossetti, M.D., Smith, J.S., Peters, B.A. (Eds.), *Proceedings of the 2004 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 89–100.

Craven, P., Wahba, G. (1979). Smoothing noisy data with spline functions. *Numerische Mathematik* 31, 377–403.

de Boor, C. (1978). *A Practical Guide to Splines*. Springer-Verlag, New York.

del Castillo, E. (1997). Stopping rules for steepest ascent in experimental optimization. *Communications in Statistics: Simulation and Computation* 26, 1599–1615.

del Castillo, E., Cahya, S. (2001). A tool for computing confidence regions on the stationary point of a response surface. *The American Statistician* 55, 358–365.

Dierckx, P. (1981). An algorithm for surface fitting with spline functions. *IMA Journal of Numerical Analysis* 1, 267–283.

Dierckx, P. (1993). *Curve and Surface Fitting with Splines*. *Monographs on Numerical Analysis*. Oxford University Press.

Donohue, J.M. (1995). The use of variance reduction techniques in the estimation of simulation metamodels. In: Alexopoulos, C., Kang, K., Goldsman, D., Lilegdon, W. (Eds.), *Proceedings of the 1995 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 194–200.

Donohue, J.M., Houck, E.C., Myers, R.H. (1993a). A sequential experimental design procedure for the estimation of first- and second-order simulation metamodels. *ACM Transactions on Modeling and Computer Simulation* 3, 190–224.

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

Donohue, J.M., Houck, E.C., Myers, R.H. (1995). Simulation designs for the estimation of quadratic response surface gradients in the presence of model misspecification. *Management Science* 41, 244–262.

Draper, N.R., Guttman, I. (1986). Response surface designs in flexible regions. *Journal of the American Statistical Association* 81, 1089–1094.

Draper, N.R., Lin, D.K.J. (1990). Small response-surface designs. *Technometrics* 32 (2), 187–194.

Eubank, R.L. (1988). *Spline Smoothing and Nonparametric Regression*. Dekker, New York.

Floudas, C.A., Pardalos, P.M. (Eds.) (1996). *State of the Art in Global Optimization*. Kluwer, Dordrecht, The Netherlands.

Franke, R. (1982). Scattered data interpolation: Tests of some methods. *Mathematics of Computation* 38, 181–200.

Friedman, J.H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics* 19, 1–141.

Fu, M.C. (1994). Optimization via simulation: A review. *Annals of Operations Research* 53, 199–248.

Fu, M.C. (2001). Simulation optimization. In: Peters, B.A., Smith, J.S., Medeiros, D.J., Rohrer, M.W. (Eds.), *Proceedings of the 2001 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 53–61.

Fu, M.C. (2003). Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing* 14, 192–215.

Glover, F., Kelly, J.P., Laguna, M. (1996). New advances and applications of combining simulation and optimization. In: Charnes, J.M., Morrice, D.J., Brunner, D.T., Swain, J.J. (Eds.), *Proceedings of the 1996 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 144–152.

Grossman, I.E. (Ed.) (1996). *Global Optimization in Engineering Design*. Kluwer, Dordrecht, The Netherlands.

Hardy, R.L. (1971). Multiquadric equations of topography and other irregular surfaces. *Journal of Geophysical Research* 76, 1905–1915.

Hussain, M.F., Barton, R.R., Joshi, S.B. (2002). Metamodeling: Radial basis functions versus polynomials. *European Journal of Operational Research* 138, 142–154.

Jin, R., Chen, W., Simpson, T.W. (2000). Comparative studies of metamodeling techniques under multiple modeling criteria. In: *Proceedings of the 8th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA*. AIAA, St. Louis, MO (CD-ROM).

Joshi, S., Sherali, H.D., Tew, J.D. (1998). An enhanced response surface methodology (RSM) algorithm using gradient deflection and second-order search strategies. *Computers and Operations Research* 25, 531–541.

Kelton, W.D., Sadowski, R.P., Sturrock, D.T. (2004). *Simulation with Arena*, 3rd edition. McGraw-Hill, New York.

Khuri, A.I., Cornell, J.A. (1987). *Response Surfaces, Designs, and Analyses*. Dekker, New York.

Kleijnen, J.P.C. (1975a). A comment on Blanning's metamodel for sensitivity analysis: The regression metamodel in simulation. *Interfaces* 5, 21–23.

Kleijnen, J.P.C. (1975b). *Statistical Techniques in Simulation, part II*. Dekker, New York.

Kleijnen, J.P.C. (1987). *Statistical Tools for Simulation Practitioners*. Dekker, New York.

Kleijnen, J.P.C. (2005). An overview of the design and analysis of simulation experiments for sensitivity analysis. *European Journal of Operational Research* 164, 287–300.

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

Kleijnen, J.P.C., van Beers, W.C.M. (2004). Application-driven sequential designs for simulation experiments: Kriging metamodeling. *Journal of the Operational Research Society* 55, 876–883.

Kleijnen, J.P.C., van Beers, W.C.M. (2005). Robustness of kriging when interpolating in random simulation with heterogeneous variances: Some experiments. *European Journal of Operational Research* 165, 826–834.

Kleijnen, J.P.C., van Groenendaal, W. (1995). Two-stage versus sequential sample-size determination in regression analysis of simulation experiments. *American Journal of Mathematical and Management Sciences* 15 (1/2), 83–114.

Kolmogorov, A.N. (1961). On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition. *American Mathematical Society Translations Series 2* 17, 369–373.

Law, A.M., Kelton, W.D. (2000). *Simulation Modeling and Analysis*, 3rd edition. McGraw-Hill, New York.

Law, A., McComas, M.G. (2002). Simulation-based optimization. In: Yücesan, E., Chen, H., Snowdon, J.L., Charnes, J.M. (Eds.), *Proceedings of the 2002 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 41–44.

Másson, E., Wang, Y.-J. (1990). Introduction to computation and learning in artificial neural networks. *European Journal of Operational Research* 47, 1–28.

MATLAB (2005). Neural network toolbox for MATLAB. Available at *http://www.mathworks.com/products/neuralnet/*.

McAllister, C.D., Altuntas, B., Frank, M., Potoradi, J. (2001). Implementation of response surface methodology using variance reduction techniques in semiconductor manufacturing. In: Peters, B.A., Smith, J.S., Medeiros, D.J., Rohrer, M.W. (Eds.), *Proceedings of the 2001 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 1225–1230.

Meckesheimer, M., Booker, A.J., Barton, R.R., Simpson, T.W. (2002). Computationally inexpensive metamodel assessment strategies. *AIAA Journal* 40, 2053–2060.

Miró-Quesada, G., del Castillo, E. (2004). An enhanced recursive stopping rule for steepest ascent searches in response surface methodology. *Communications in Statistics: Simulation and Computation* 33, 201–228.

Mitchell, T.J., Morris, M.D. (1992). The spatial correlation function approach to response surface estimation. In: Swain, J.J., Goldsman, D., Crain, R.C., Wilson, J.R. (Eds.), *Proceedings of the 1992 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 565–571.

Montgomery, D.C. (2001). *Design and Analysis of Experiments*, 5th edition. Wiley, New York.

Morris, M.D. (2000). A class of three-level experimental designs for response surface modeling. *Technometrics* 42, 111–121.

Myers, R.H. (1999). Response surface methodology – Current status and future directions. *Journal of Quality Technology* 31, 30–44.

Myers, R.H., Khuri, A.I. (1979). A new procedure for steepest ascent. *Communications in Statistics: Theory and Methods* 8, 1359–1376.

Myers, R.H., Montgomery, D.C. (2002). *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, 2nd edition. Wiley, New York.

Neddermeijer, H.G., van Oortmarssen, G.J., Piersma, N., Dekker, R. (2000). A framework for response surface methodology for simulation optimization. In: Joines, J.A., Barton, R.R., Kang, K., Fishwick, P.A. (Eds.), *Proceedings of the 2000 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 129–136.

Netlab (2005). How to use Netlab. Available at *http://homepages.cae.wisc.edu/~ece539/software/netlab/intro.htm*.

NETLIB (2005). DIERCKX: Fortran subroutines for calculating smoothing splines for various kinds of data and geometries, with automatic knot selection. Available at *http://www.netlib.org/dierckx/*.

Nicolai, R.P., Dekker, R., Piersma, N., van Oortmarssen, G.J. (2004). Automated response surface methodology for stochastic optimization models with unknown variance. In: Ingalls, R.G., Rossetti, M.D., Smith, J.S., Peters, B.A. (Eds.), *Proceedings of the 2004 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 491–499.

Nozari, A., Arnold, S.F., Pegden, C.D. (1987). Statistical analysis for use with the Schruben and Margolin correlation induction strategy. *Operations Research* 35, 127–139.

Peterson, J.J., Cahya, S., del Castillo, E. (2002). A general approach to confidence regions for optimal factor levels of response surfaces. *Biometrics* 58, 422–431.

Ramberg, J.S., Sanchez, S.M., Sanchez, P.J., Hollick, L.J. (1991). Designing simulation experiments: Taguchi methods and response surface metamodels. In: Nelson, B.L., Kelton, W.D., Clark, G.M. (Eds.), *Proceedings of the 1991 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 167–176.

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

Sanchez, S. (2000). Robust design: Seeking the best of all possible worlds. In: Joines, J.A., Barton, R.R., Kang, K., Fishwick, P.A. (Eds.), *Proceedings of the 2000 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 69–76.

Santner, T.J., Williams, B.J., Notz, W.I. (2000). Sequential design of computer experiments to minimize integrated response functions. *Statistica Sinica* 10, 1133–1152.

Santner, T.J., Williams, B.J., Notz, W.I. (2003). *The Design and Analysis of Computer Experiments*. Spinger-Verlag, New York. PeRK code available at *http://www.stat.ohio-state.edu/~comp_exp/index.html*, June 30, 2005.

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

Shao, J. (1993). Linear model selection by cross-validation. *Journal of the American Statistical Association* 88, 486–494.

Shin, M., Sargent, R.G., Goel, A.L. (2002). Gaussian radial basis functions for simulation metamodeling. In: Yücesan, E., Chen, C.-H., Snowdon, J.L., Charnes, J.M. (Eds.), *Proceedings of the 2002 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 483–488.

Simpson, T.W., Mauery, T.M., Korte, J.J., Mistree, F. (1998). Comparison of response surface and kriging models for multidisciplinary design optimization. In: *7th Symposium on Multidisciplinary Analysis and Optimization*. *AIAA-98-4755*. AIAA, St. Louis, MO, pp. 381–391.

Simpson, T.W., Frecker, M., Rothrock, L., Stump, G., Barton, R.R., Ligetti, C. (2003). Assessing the impact of graphical design interfaces on design efficiency and effectiveness. *ASME Journal of Computer and Information Science and Engineering* 3, 144–154.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions (with discussion). *Journal of the Royal Statistical Society B* 36, 111–133.

Tew, J.D., Wilson, J.R. (1992). Validation of simulation analysis methods for the Schruben–Margolin correlation-induction strategy. *Operations Research* 40, 87–103.

Tew, J.D., Wilson, J.R. (1994). Estimating simulation metamodels using combined correlation-based variance reduction techniques. *IIE Transactions* 26, 2–16.

Thurston, D.L., Carnahan, J.V., Liu, T. (1994). Optimization of design utility. *ASME Journal of Mechanical Design* 116, 801–807.

Tibshirani, R. (1996). A comparison of some error estimates for neural network models. *Neural Computation* 8, 152–163.

Tomick, J., Arnold, S., Barton, R.R. (1995). Sample size selection for improved Nelder–Mead performance. In: Alexopolous, C., Kang, K., Lilegdon, W.R., Goldsman, D. (Eds.), *Proceedings of the 1995 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 341–345.

Tu, C.-H., Barton, R.R. (1997). Production yield estimation by the metamodel method with a boundary-focused experiment design. In: *Proceedings of DETC'97, 1997 ASME Design Engineering Technical Conference*. *DETC97/DTM3870*. ASME, Fairfield, NJ.

van Beers, W.C.M., Kleijnen, J.P.C. (2004). Customized sequential designs for random simulation experiments: Kriging metamodeling and bootstrapping. CentER Discussion Paper No. 2004-63, Tilburg University.

Watlington, J.A. (2005). Polynomial and radial basis function example code. Available at *http://web.media.mit.edu/~wad/mas864/src/polynomial.html*.

Yesilyurt, S., Patera, A.T. (1995). Surrogates for numerical simulations; optimization of eddy-promoter heat exchangers. *Computer Methods in Applied Mechanics and Engineering* 121, 231–257.

Zionts, S. (1992). The state of multiple criteria decision making: Past, present, and future. In: Goicoechea, A., Duckstein, L., Zionts, S. (Eds.), *Multiple Criteria Decision Making*. Springer-Verlag, New York, pp. 33–43.