Journal of Global Optimization (2005) 33: 31–59
DOI 10.1007/s10898-004-6733-1

On the Design of Optimization Strategies Based on Global Response Surface Approximation Models

ANDRÁS SÓBESTER[1], STEPHEN J. LEARY[2] and ANDY J. KEANE[3]
[1] *University of Southampton, School of Engineering Sciences, Highfield, Southampton SO17 1BJ, UK (e-mail: a.sobester@soton.ac.uk)*
[2] *University of Southampton, School of Engineering Sciences, Highfield, Southampton SO17 1BJ, UK (e-mail: s.j.leary@soton.ac.uk)*
[3] *University of Southampton, School of Engineering Sciences, Highfield, Southampton SO17 1BJ, UK (e-mail: ajk@soton.ac.uk)*

(Received 10 March 2003; accepted 4 November 2004)

Abstract. Striking the correct balance between global exploration of search spaces and local exploitation of promising basins of attraction is one of the principal concerns in the design of global optimization algorithms. This is true in the case of techniques based on global response surface approximation models as well. After constructing such a model using some initial database of designs it is far from obvious how to select further points to examine so that the appropriate mix of exploration and exploitation is achieved. In this paper we propose a selection criterion based on the expected improvement measure, which allows relatively precise control of the scope of the search. We investigate its behavior through a set of artificial test functions and two structural optimization problems. We also look at another aspect of setting up search heuristics of this type: the choice of the size of the database that the initial approximation is built upon.

Key words: expected improvement, Gaussian kernels, Radial basis functions

1. Introduction

The rapid advances seen in recent times in computing technology have brought about changes in many facets of computational engineering. One aspect that has not changed, however, is the relative computational expense of simulations used in design optimization. The evaluation of physics-based models can still take tens of hours and, as the demand for ever-higher fidelity (and thus more complex) models closely shadows increases in processing power, this is unlikely to change in the near future.

One of the more popular approaches for the economical use of such expensive simulations is the broadening class of optimization algorithms based on cheap global approximation models (often called surrogate models or spatial prediction models) of the high-fidelity computational simulation. These models involve running the physics-based analysis code (treating it, essentially, as a black-box function) for a number of designs and using this training data to build a surrogate model, which is cheap to evaluate. These

models have their roots in a variety of fields: response surface approximation methodology (low-order polynomials), nature-inspired computing (artificial neural networks), spatial statistics, stochastic process theory, mathematical geology (kriging, Radial Basis Function models), etc.

A detailed taxonomy of optimization methods based on global approximation models is provided by Jones (2001). In this study we limit ourselves to two-stage procedures, which are based on the following general template. First, an initial set of sample points is generated using some Design of Experiments (DoE) technique. Generally, at this stage the location of the points is only required to satisfy some space-filling criterion. The situation may be slightly different if the design space is constraint-bound and the computational cost of the constraint is relatively low: if we expect the best designs to lie on the boundaries, we can place some of the initial points on the boundary. Also, the constraints may be used to trim the bounding box. Nevertheless, in most cases uniform distribution of the points is the best we can do, as we have not computed any objective function values yet and thus have no knowledge of areas of interest on the landscape under scrutiny. We then run the simulation for these designs and build the initial surrogate model.

The second stage of the method is the selection of the so-called *infill sample*, i.e., the next point(s) to be evaluated, followed by the reconstruction of the approximation model (this process is then repeated until we run out of time). When selecting the infill sample we can base our choice on information gleaned from the current approximation. The simplest infill selection criterion is the predictor itself: we can optimize the current approximation (of course, this is a cheap operation, as no further calls to the physics-based analysis are required) and sample at the optimum found. Another possible strategy is to find the point where the estimated error of the predictor is at a maximum (such a measure is available, for example, in polynomial, kriging and Gaussian Radial Basis Function models), i.e. where we are least certain about the predicted objective value. Thus, the next re-fit of the surrogate model will yield a prediction with a more uniform global accuracy. As we shall see later, both of these criteria have their drawbacks and we investigate some more sophisticated alternatives.

For now, let us return to the first stage of our generic optimizer: the construction of the initial approximation. There are a wide variety of DoE methods available to the designer wishing to select the initial sample points. Factorial, fractional factorial, central composite (Montgomery, 2000), latin hypercube (Mackay et al., 1979) and LP_τ (Sobol, 1979) designs are amongst the most widely used. The goal of these techniques is to fill the design space in some sense, because, as we mentioned earlier, it is commonly recognized that in the absence of any *a priori* knowledge of the

problem under consideration (such as the location of a constraint boundary), uniformity of the design points throughout the domain is favorable. Our main concern in this study is the *optimum size* of this initial sample. Although some researchers use "rules of thumb", such as the number of points should be roughly ten times the number of dimensions (Jones et al., 1998), to date there is no clear understanding of how this figure should be chosen and what influence the choice has on the performance of the optimizer.

Intuitively, one might keep the size of the initial sample to a minimum and target the majority of the available shots more intelligently, i.e., using some infill sample selection criterion based on an approximation of the objective function. However, caution needs to be exercised here. The main question is: could the approximation based on a very small sample be so inaccurate – and thus misleading – that we would be better off starting with a set of points whose selection is simply based on a space-filling criterion? Conversely, if we start with a large number of data points, are we not wasting precious evaluations by selecting them without regard to the previously found objective values? Also, how (if at all) does the optimum size of the initial sample depend on the choice of the type of infill criterion for further points? While a definitive answer may be some way away, this paper presents an empirical investigation of the issue, sufficiently conclusive to offer some set-up guidance to users of such algorithms.

The second object of our study, which we examine in conjunction with the problem of the initial sample size, concerns stage two of the approximation-based search. We look at how the scope of the infill criterion can be controlled, i.e., how it can be biased towards local exploitation of promising basins of attraction or towards global exploration of the search space and, most importantly, what effect the bias has on the performance of the optimizer.

As we mentioned earlier, it is possible to select the next sample point simply by optimizing the predictor. This may work well on simple, unimodal functions, which can be approximated well even with a relatively small number of points, but it may easily get trapped in a local optimum if the landscape is multimodal. The other extreme is to always choose the point where the uncertainty associated with the predictor is highest. This procedure has the merit of guaranteeing global convergence (under certain rather mild assumptions), but it may require a very large number of evaluations to achieve this even on simple problems. Watson and Barnes (1995) suggest sampling in threshold-bounded extremes, i.e., to find the next point by maximizing the probability of the infill sample objective value exceeding some threshold. The drawback here, as Sasena et al. (2002) point out, is similar to that of searching the predictor: the optimization will become extremely localized and thus prone to premature stall.

In global optimizers it is important to achieve a balance between exploration and exploitation – approximation-based techniques are no exception. An infill selection criterion designed to search both the prediction and its uncertainty is the maximization of the expectation of the amount by which the next potential evaluated point will improve on the best objective value known so far. This figure of merit is often termed *expected improvement*. The concept goes back at least to Mockus et al. (1978) and has recently been used for optimization by several researchers, most notably by Jones et al. (1998) in their EGO algorithm. By always placing the next sample point at the maximum of the expected improvement landscape associated with the approximation of the current model, the search is likely to visit promising basins of attraction, while occasionally sampling in other, less well mapped areas of the search space as well. Another advantage of the expected improvement measure is that optimization strategies based on it, using the template described earlier, can be implemented on parallel architectures. In this case, the top N_p local maxima of the expected improvement are highlighted as the next infill sample (the expected improvement surface is almost invariably multimodal), where N_p is the number of available processors. It has been shown that the parallel speedup is often close to linear when using this approach (Sóbester et al., 2004)).

Thus, with expected improvement we have a means of fusing exploration and exploitation into a single criterion. However, if the problem in hand is likely to yield a simple, unimodal surface, searching the predictor will probably work better. Conversely, if the objective landscape is extremely multimodal, biasing the search towards sampling in thus far unexplored areas could lead to faster convergence than the expected improvement criterion. In other words, expected improvement still does not allow us to *control* the balance between local and global exploration. Furthermore, the scope of the expected improvement criterion may not be broad enough if the objective is poorly estimated by the approximation (and consequently the accuracy of the expectation of the improvement is also questionable). To alleviate these shortcomings, Schonlau (1997) proposes the *generalized expected improvement* criterion. This is controlled by a parameter $g = 0, 1, 2, \ldots$ He shows that for $g = 0$ the criterion yields the probability of improvement (this value, for a particular point, is the probability of the current best objective value being improved on if we sample in that point). For higher values of g the emphasis shifts more and more towards global search. Sasena et al. (2002) suggest a heuristic reminiscent of Simulated Annealing, which is based on generalized expected improvement. They start with a high value of g, which is then decreased as the search progresses, based on a discrete, approximately exponential cooling schedule. The generalized expected improvement measure thus allows the user to control the scope of the search to some extent, but since it has no upper bound, its

values are extremely difficult to select for a particular application (it is hard to tell how much impact a change from, say, $g=5$ to, say, $g=10$ will make on the global bias of the search). Furthermore, it does not cover the search scope range between extremely localized exploitation ($g=0$, i.e., probability of improvement) and expected improvement ($g=1$).

In this paper we propose a weighted expected improvement criterion, which is designed to allow a more flexible and more "user friendly" means of biasing the search towards exploration or exploitation. The next section describes this measure in more detail, taking the standard expected improvement criterion as a natural starting point. This is followed by a demonstrative example (Section 3) that highlights the most important features of the criterion. Section 4 deals with the question of how to perform weighted expected improvement updates on constrained landscapes. We then adopt an empirical approach to examine the effect of the weighting (the control parameter of the criterion) on the performance of an approximation-based optimizer, in conjunction with the other major parameter that we propose to investigate: the size of the initial sample. The first part of Section 5 presents results obtained on a set of artificial test functions, while in the second and third parts we use two structural optimization problems to verify some of the conclusions gleaned from these results. We then discuss a variable-bias implementation of the weighted expected improvement criterion and we compare its performance to that of some well-known techniques. In Section 7 we summarize our conclusions and suggests pointers to further work.

2. Radial basis function interpolators and weighted expected improvement

Before delving into the details of our proposed infill selection criterion, we briefly review the background of stage one of the optimizer, i.e., the construction of the global approximation of the objective function. This surrogate model can be built as soon as we have chosen a suitable experimental design and evaluated the high fidelity model at this set of inputs. In a typical approximation model the relationship between observations (responses) and independent variables on a k-dimensional domain D is expressed as

$$y = f(\mathbf{x}), \tag{1}$$

where y is the observed response, $\mathbf{x}$ is a vector of k independent variables

$$\mathbf{x} = (x_1, x_2, \ldots, x_k) \tag{2}$$

and $f(\mathbf{x})$ is some unknown function. An approximation to this response

$$\hat{y} = \hat{f}(\mathbf{x}) \tag{3}$$

is sought. As we hinted earlier, it is also important from the optimization point of view to be able to obtain an error estimate for this approximation. To this end, here we employ a stochastic process-based modeling framework. This may seem counterintuitive, as physics-based numerical computer experiments, the pillars of the majority of computer-aided design optimization procedures, are usually deterministic. That is, unlike physical experiments, repeated runs of such simulations on the same design return the same figure of merit (objective value) each time. Nevertheless, it can be argued that stochastic process approximation techniques can be employed to model this type of output. The rationale is that, although the physics-based simulation process itself is deterministic, its output can be viewed as *a realisation of a stochastic process*.[1]

There are several approaches for building such models – here we choose to work with Radial Basis Functions (RBF) on the grounds that their training is inexpensive, yet, as we will see, they are sufficiently accurate for optimization purposes. RBF models attempt to express a complicated landscape as the weighted sum of several simple functions – in the following we describe the model building procedure in more detail.

Assuming that we can afford to run the analysis code N times, we sample the objective for N designs (denoted by $[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}]$), at which we obtain the responses $\mathbf{y} = [y^{(1)}, y^{(2)}, \ldots, y^{(N)}]$. The RBF can be used to make a prediction $\hat{y} = \hat{f}(\mathbf{x})$ at any point $\mathbf{x}$ in the design space and the first step towards this is to choose the basis function centres. To obtain an interpolating model we need at least N bases. The common choice here is the set of N points where we know the objective function values (i.e., $[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}]$). The basis functions take the form $\phi(\|\mathbf{x} - \mathbf{x}^{(i)}\|)$, where $\phi(\cdot)$ is some (usually) non-linear function, the ith such function depending on the Euclidean distance between $\mathbf{x}$ and $\mathbf{x}^{(i)}$. The predictor is a linear combination of these basis functions, that is

$$\hat{y} = \hat{f}(\mathbf{x}) = \sum_{i=1}^{N} w_i \phi(\|\mathbf{x} - \mathbf{x}^{(i)}\|). \tag{4}$$

The coefficients w_i have to be found such that the predictor interpolates the data. To do this, we are required to satisfy for $j = 1, \ldots, N$

[1] We note here that there is some debate concerning the validity of this fiction and statisticians are sometimes reluctant to interpret predictors and error measures derived from it as more than practically useful figures, suggesting that no pretense should be made about the rigorousness of their mathematical foundation. The idea is nonetheless a powerful one, as it suggests plausible ways of constructing useful models of deterministic outputs (Trosset and Torczon (1997)) and experience shows that the predictions obtained with them are adequate for practical purposes.

$$\hat{f}(\mathbf{x}^{(j)}) = \sum_{i=1}^{N} w_i \phi(\|\mathbf{x}^{(j)} - \mathbf{x}^{(i)}\|) = y^{(j)}. \tag{5}$$

We note here that some authors also add a set of polynomial terms to the expression of the predictor in order to guarantee that the system of equations will not be singular (see, e.g., Jones (2001)). In our experience this rarely appears to be necessary, so, for the sake of simplicity, we have omitted it here.

Defining the coefficient vector $\mathbf{w} = [w_1, w_2, \ldots, w_N]^T$ and the matrix $\mathbf{\Phi}_{i,j} = \phi(\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|)$, where $i = 1, \ldots, N$ and $j = 1, \ldots, N$, Equation (5) can be written as $\mathbf{\Phi w} = \mathbf{y}^T$. Then, provided the inverse of $\mathbf{\Phi}$ exists, the coefficients can be determined by computing $\mathbf{w} = \mathbf{\Phi}^{-1}\mathbf{y}^T$ and a prediction $\hat{y}$ can be made in any point $\mathbf{x}^{(N+1)} \in D$

$$\hat{y}^{(N+1)} = \boldsymbol{\phi}\mathbf{w} = \boldsymbol{\phi}\mathbf{\Phi}^{-1}\mathbf{y}^T, \tag{6}$$

where

$$\boldsymbol{\phi} = \left[\phi\left(\|\mathbf{x}^{(N+1)} - \mathbf{x}^{(1)}\|\right), \phi\left(\|\mathbf{x}^{(N+1)} - \mathbf{x}^{(2)}\|\right), \ldots, \phi\left(\|\mathbf{x}^{(N+1)} - \mathbf{x}^{(N)}\|\right)\right]. \tag{7}$$

Many different basis functions $\phi(\cdot)$ could be considered. Throughout this work we have used exponentially decaying Gaussian basis functions

$$\phi(r) = \exp\left(\frac{-r^2}{2\sigma^2}\right) \tag{8}$$

as they facilitate the derivation of an expected improvement measure. The choice of the hyperparameter σ, which governs the regions of influence of each kernel, is important and can affect prediction accuracy. In the work presented here we use a *leave-one-out cross validation* procedure, searching for the optimum σ over the domain $[10^{-2}, 10^{1}]$. This means that for each value of σ we build N RBF models leaving out one of the training points in each case (as though we only had $N-1$ points), we compute the difference between the true objective value of the currently left out point and the objective predicted by the partial model (which uses the remaining $N-1$ points) at the same point. The final model is constructed using the σ that minimizes the sum of the squares of these residuals. We consider 20 values of σ logarithmically spread over the range indicated above. More thorough searches (i.e., a full optimization of the hyperparameter σ) could be considered, but this would increase the computational cost of the training and in the authors' experience the gain in model accuracy thus achieved is not significant. We also note here that the optimum σ is related to the distances between the kernels – the range $[10^{-2}, 10^{1}]$ appears to be suitable for the

case when the problem domain is normalized to $D = [0, 1]^k$ (as in all the experiments described here).

Clearly, more accurate models could be considered; for example, we could allow the selection of a different σ for each independent variable (as is often done, for example, in kriging). However, the computational cost of model training then becomes an issue.

We mentioned in the introduction that the global end of the search strategy spectrum is to sample in areas of high estimated approximation error, i.e., in our case, to choose the design that maximizes the estimated error of the RBF predictor. In order to calculate this we make use of the assumption discussed earlier, namely that each deterministic response $y(\mathbf{x})$ is in fact the realisation of some stochastic process $Y(\mathbf{x})$ (taken here to be a Gaussian random variable). Using the (Gaussian) distributions of the N responses $\mathbf{y} = [y^{(1)}, y^{(2)}, \ldots, y^{(N)}]$ collected so far, it can be shown that the mean and the variance of the assumed stochastic process at $\mathbf{x}^{(N+1)}$ are

$$\hat{y}^{(N+1)} = \boldsymbol{\phi}\Phi^{-1}\mathbf{y}^T, \tag{9}$$

$$\sigma^2_{\hat{y}^{(N+1)}} = 1 - \boldsymbol{\phi}\Phi^{-1}\boldsymbol{\phi}^T, \tag{10}$$

respectively (for a detailed demonstration see, e.g., Gibbs (1997)). As expected, the mean of the imaginary Gaussian distribution that we drew $\hat{y}^{(N+1)}$ from (Equation (9)) is, in fact, the RBF predictor obtained earlier (6). We will use the variance of this Gaussian distribution (Equation (10)) as a measure of the likely prediction error at untested sites.

From a global optimization perspective, searching the prediction error amounts to *exploration* of the search space, whereas searching the predictor itself is equivalent to *exploiting* currently known promising basins of attraction. Clearly, we need an infill-point selection criterion that balances these two approaches.

As we have seen, the stochastic process $Y(\mathbf{x})$ models our uncertainty about the response $y(\mathbf{x})$ in the point $\mathbf{x}$. Denoting the best objective value from the sample evaluated so far by $y_{\min} = \min\left\{y^{(1)}, y^{(2)}, \ldots, y^{(N)}\right\}$, a further quantity can be defined: the *improvement*

$$I(\mathbf{x}) = \max\{y_{\min} - Y(\mathbf{x}), 0\}. \tag{11}$$

(Jones et al., 1998). This is, of course, also a random variable – it models our uncertainty about the amount by which $y(\mathbf{x})$, the objective function value in the next evaluated sample point, will improve on the current best objective.

Given a prediction $\hat{y}$ and an error estimate $s = \sigma_{\hat{y}^{(N+1)}}$ (in a point $\mathbf{x}$, as per Equations (9) and (10)), using Gaussian kernels, the expectation of

the improvement (or, as it is often termed in the literature, the *expected improvement*) can be calculated (see, e.g., Schonlau, 1997) as

$$E(I) = EIF(\mathbf{x}) = \begin{cases} (y_{\min} - \hat{y})\Psi\left(\frac{y_{\min}-\hat{y}}{s}\right) + s\psi\left(\frac{y_{\min}-\hat{y}}{s}\right) & \text{if } s > 0, \\ 0 & \text{if } s = 0, \end{cases} \quad (12)$$

where $\Psi(\cdot)$ is the standard normal distribution function and $\psi(\cdot)$ is the standard normal density function.

The first term of Equation (12) is the predicted difference between the current minimum and the prediction $\hat{y}$ in $\mathbf{x}$, penalized by the probability of improvement. Hence it is large where $\hat{y}$ is small (or it is likely to be smaller then $y_{\min}$). The second term is large when the error s is large, i.e., when there is much uncertainty about whether y will be better than $y_{\min}$. Thus, as Schonlau (1997) points out, the expected improvement will tend to be large at a point with predicted value smaller than $y_{\min}$ and/or there is much uncertainty associated with the prediction. Therefore, expected improvement can be considered as a balance between seeking promising areas of the design space (according to our approximation) and the uncertainty in the model. The global search strategy based on it (i.e., evaluation of the initial DoE set, followed by updates at maxima of the expected improvement surface) has the advantage that it is much less likely to stall than a search over the approximation only (although there are certain pathological cases when it does, see, e.g., Jones (2001)). The disadvantage is that it usually takes longer to converge, which could be a drawback if the initial model *did* turn out to be an accurate one.

Since we are interested in controlling the precise balance of exploitation (optimization of the predictor) and exploration (seeking areas of maximum uncertainty), it makes sense to introduce a weighted infill sample criterion, which is a linear combination of the two terms of the expected improvement measure

$$WEIF(\mathbf{x}) = \begin{cases} w(y_{\min} - \hat{y})\Psi\left(\frac{y_{\min}-\hat{y}}{s}\right) + (1-w)s\psi\left(\frac{y_{\min}-\hat{y}}{s}\right) & \text{if } s > 0, \\ 0 & \text{if } s = 0, \end{cases} \quad (13)$$

where the weighting factor $w \in [0, 1]$. Clearly, $w = 0$ will yield the global extreme of the search scope range, while selecting the next infill sample point using $w = 1$ will concentrate the search on the current best basin of attraction. Thus, the larger the values of w, the more restricted (local) the scope of the search will be and the weighting offers the possibility of fully covering the continuum between exploration and exploitation. A notable value of w is 0.5, which will, of course, yield $0.5EIF(\mathbf{x})$. We now examine the impact of varying the weighting w on the Weighted Expected

Improvement Function (WEIF) landscape through a simple one-variable toy problem.

3. A demonstrative example

Let us consider the one-variable function shown at the top of Figure 1. We assume that it has been sampled in the six points indicated on the plot as circles and an RBF approximation has been constructed. The predictor is also shown in the top section of the figure. The dashed lines either side of the predictor represent the predictor plus and minus one standard error. On the left-hand side of the plot this is very small, as the density of the sample is much larger there – the error is only visible in the fairly large gap between the fifth and sixth points.

The uneven distribution of these points may seem slightly unrealistic in a DoE context. Nevertheless, such situations regularly occur in higher dimensions and/or after a few infill points have been added to the database.

The section of the figure placed below this plot shows the weighted expected improvement with weighting $w=0$. Based on the above discussion,

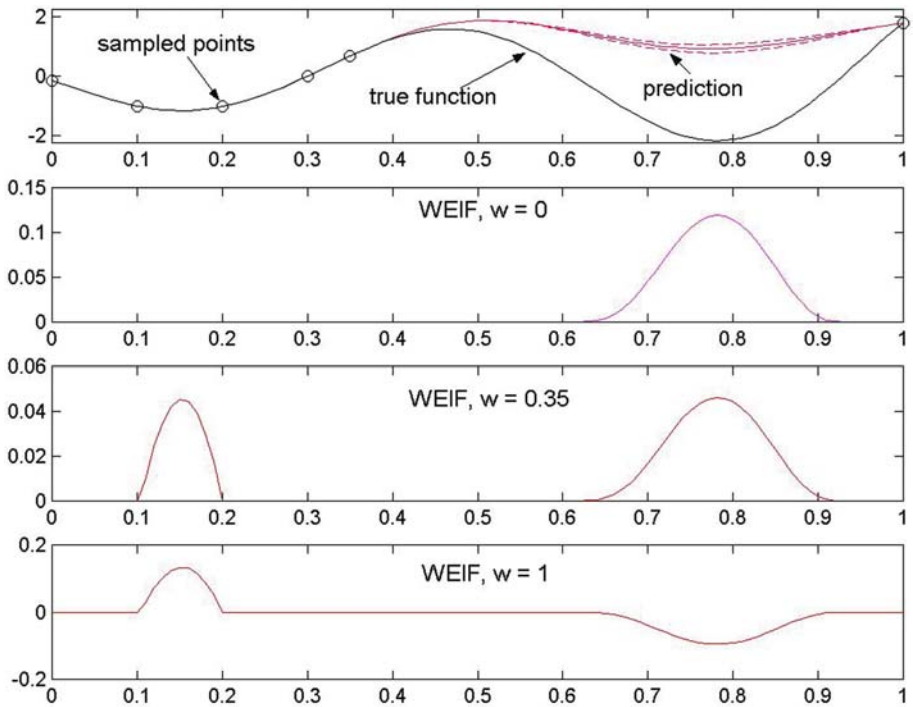

Figure 1. Demonstrative example showing the impact of the choice of w on the weighted expected improvement for a simple one-variable function.

we would expect the WEIF measure to give the search a fully global bias. This is indeed the case, as the sole maximum of the criterion is in the sparsely sampled region, where the uncertainty about whether the function value is better than the current best point is high. Note that the "goodness" of the prediction, i.e., the value of the predictor, is not influencing the optimization based on the WEIF with $w = 0$: the weighted criterion guides us to near the middle of the unsampled region, in spite of the predicted objective being rather poor here.

As we increase w, i.e., we give the search a more local flavor, another peak starts to emerge in WEIF on the left-hand side where the predictor indicates good function values (although the uncertainty is very low here). When we reach $w = 0.35$ (see the third section of Figure 1) the importance of exploration and exploitation becomes approximately equal (the two peaks are of equal height).

The bottom section of Figure 1 shows the WEIF landscape for $w = 1$. Clearly, maximizing this surface will yield a single optimum, which will be in the lowest predicted objective function value point. Wherever the prediction is worse than the current best point, the WEIF will be negative (due to the $y_{\min} - \hat{y}$ factor in the first term of Equation (13)).

A final aspect of our generic two-stage optimization strategy, which we need to discuss before looking at more complex examples, is the way in which we handle constraints imposed on the objective function – we do this in the following section.

4. Dealing with constrained objectives

Constrained expensive optimization problems come in two flavors. The objective is either constrained by a function that can be evaluated at negligible cost (this usually means that it can be calculated using some closed form expression) and thus it is known exactly throughout its domain (we present an engineering design example for this in Section 5) or by another computationally expensive function, in which case this, just like the objective, needs to be approximated.

The simplest constrained optimization strategy is to evaluate the constraint at the same points where the objective is evaluated and modify the expected improvement criterion to take into account the constraint values. One method for modifying the EI criterion, termed *expected violation*, is discussed by Audet et al. (2000). Another approach, used by Jones et al. (1998), involves multiplying the expected improvement criterion by an estimate of the probability that the sampled point will be feasible. Here we modify the EI criterion in a manner that, though very straightforward, is sufficiently effective to allow initial explorations on how weighted EI

might perform on constrained problems.[2] We simply set the criterion to zero wherever the approximate or exact constraints are violated

$$WEIF(\mathbf{x}) = \begin{cases} w(y_{\min} - \hat{y})\Psi\left(\frac{y_{\min}-\hat{y}}{s}\right) + (1-w)s\psi\left(\frac{y_{\min}-\hat{y}}{s}\right) \\ \text{if } s > 0 \text{ and the (approximate or exact) constraints are satisfied,} \\ 0 \\ \text{if } s = 0 \text{ or if the (approximate or exact) constraints are violated.} \end{cases} \tag{14}$$

Here $y_{\min}$ should be taken as the minimum *feasible* response (assuming, of course, that the initial sample contains at least one feasible point[3]), where feasibility is assessed on the basis of the exactly known or the approximate constraint value in each sampled point, depending on the cost of the constraint.

We are now ready to proceed to examine the effects of local/global bias of the WEIF, in conjunction with that of starting from different initial samples, on the performance of a two-stage optimization algorithm. We use artificial test functions first, followed by two "real-life" engineering applications.

5. Empirical results

5.1. ARTIFICIAL TEST FUNCTIONS

For the purposes of this study we have selected three test functions. In order of increasing complexity they are the Sphere, a modified version of Rosenbrock's "banana" function and the highly multimodal Ackley function (Ackley, 1987). Figures 2–4 represent these functions in two dimensions (please refer to the captions for details on their definition) – we have looked at the performance of the WEIF-based optimizer on the 5-variable versions and in one case (Ackley's) on the 10-dimensional landscape as well.

The optimization algorithm we have used to perform this empirical study is shown in Figure 5. We start by generating an initial database of points and we evaluate their objective function values. A random Latin Hypercube

[2]Difficulties could arise here when the optimum is on a constraint boundary. There is a rich literature on how to tackle this problem within various optimization algorithms – the interested reader may consult, for example, the seminal work of Fiacco and McCormick (1968).

[3]If the initial sample does *not* contain a feasible point, one can start by applying the algorithm to the sum of squared constraint violations instead of to the objective function. Once a feasible point has been found, one can apply the algorithm to minimize the objective. Note that the initial sample would now be augmented to include the points evaluated while searching for a feasible point.

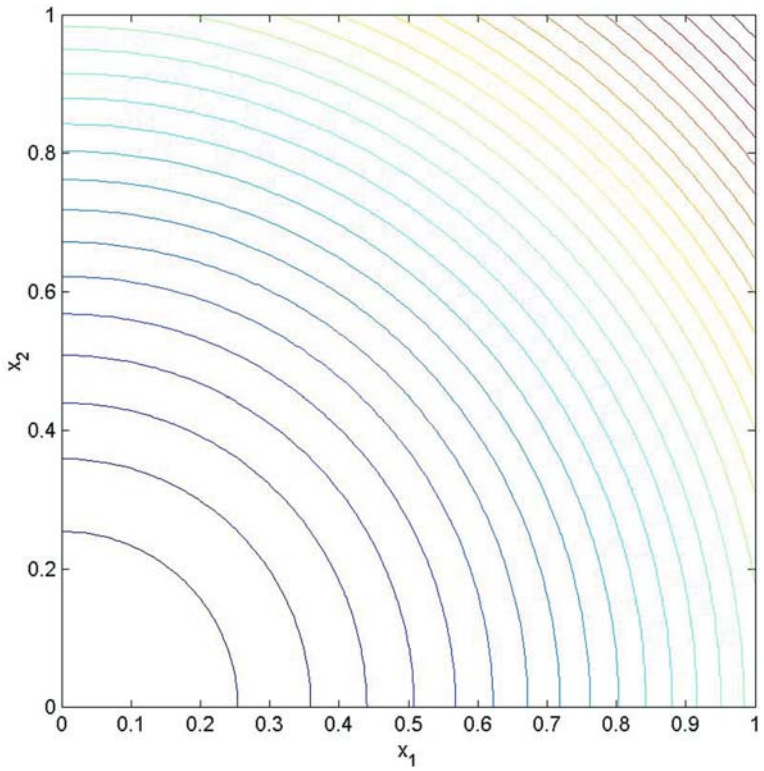

Figure 2. Sphere function.
$f_S = \sum_{i=1}^{N} x_i^2, x_i \in [0, 1].$

experimental design is used to build these initial designs. In order to alleviate the effects of any bias due to some of these initial points falling, by sheer luck, close to the global optima of the studied functions, each result is averaged over 30 runs (except where otherwise stated).

Next, an RBF model (Equation (4)) is fitted using Gaussian basis functions (Equation (8)) and the corresponding WEIF surface is optimized using a BFGS search with 1000 random restarts (like the expected improvement, the weighted expected improvement surface can be highly multimodal and thus difficult to optimize reliably – hence the large number of restarts). The objective function is evaluated at the optimum point and it is added to the database. A new RBF model is fitted and the process is repeated, usually until we run out of time. There are two exceptions to this stopping criterion, which can halt the process earlier. First, in the case of test functions with known optima, if the global optimum is reached the process stops. The second supplementary stopping criterion is employed when the WEIF weighting is very high, so that the search is becoming so localized that successive points are very close together and there is no point in continuing.

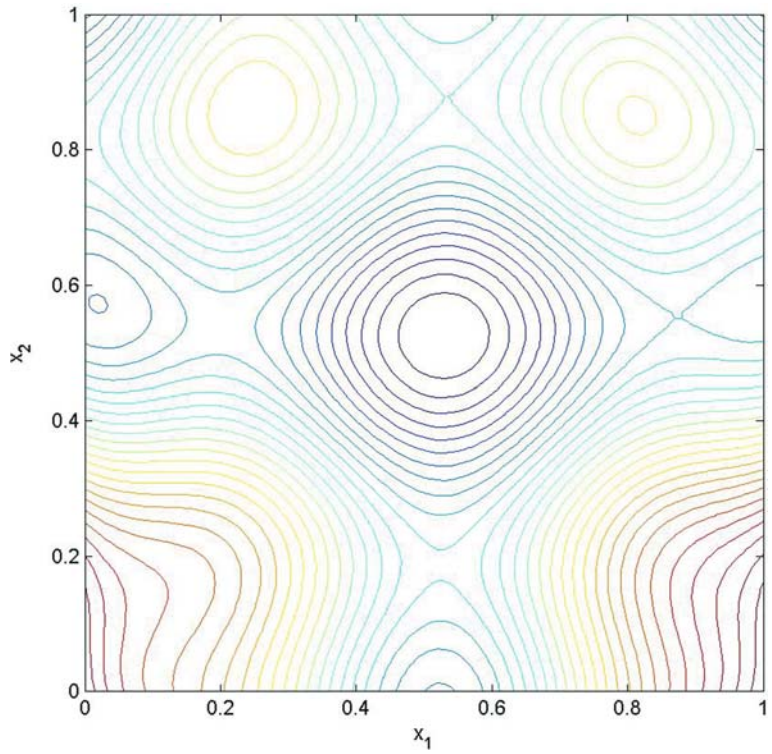

Figure 3. Modified Rosenbrock function, normalized to [0,1] (a k-dimensional sinewave has been added to the Rosenbrock function to increase its modality).

$$f_{MR} = \sum_{i=1}^{k-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2 + \sum_{i=1}^{k} 75\sin(5(1 - x_i)), \quad x_i \in [-2.048, 2.048].$$

For the visualization of the results we have opted for a greyscale snapshot map format. The density of a particular region of the plot represents the (averaged) objective function value reached by an optimizer started from an initial Latin Hypercube sample of the size shown on the vertical axis, using the WEIF weighting indicated by the horizontal axis. Looking back at the discussion of the relationship between the scope of the search and the WEIF weighting, the closer we are to the left edge of the plot the more global the search is – conversely, moving to the right gradually reduces the scope of the WEIF criterion.

Let us use Figure 6 to clarify how the plots are structured (we will analyze its actual significance later). As an example, the density of the point marked by the cross (we chose this point arbitrarily) indicates the average objective value obtained when optimizing the function under scrutiny, after the evaluation of 11 points in total (as indicated by the title of the plot), out of which five were in the initial Latin Hypercube DoE set (this value can be read off the vertical axis) and the remaining six have been

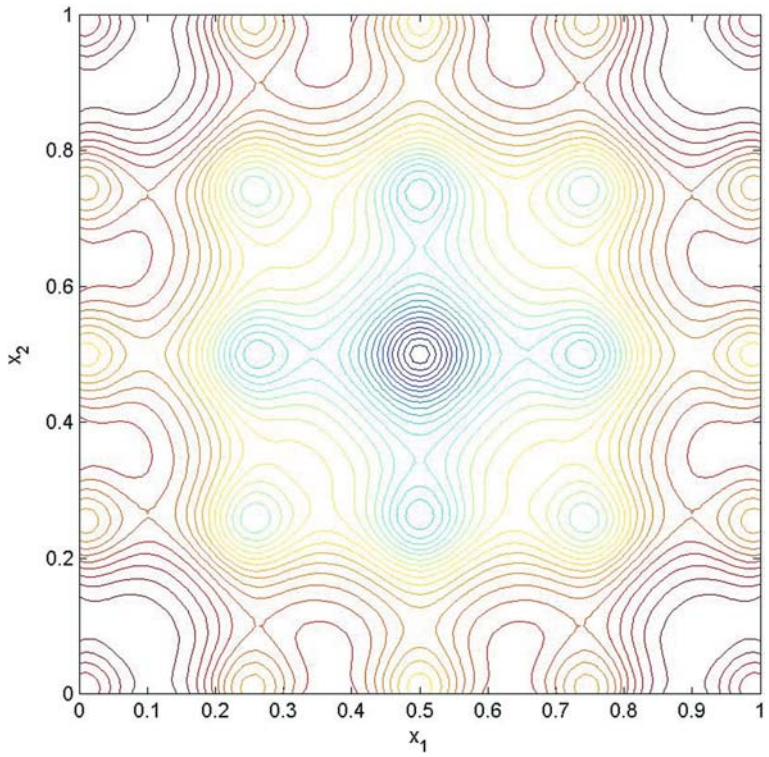

Figure 4. Ackley's path function, normalized to [0,1].

$$f_{ACK} = -a\exp\left(-b\sqrt{\frac{1}{n}\sum_{i=1}^{k} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{k}\cos(cx_i)\right) + a + \exp(1), \quad x_i \in [-2.048, 2.048].$$

selected using the WEIF criterion with a weighting of 0.4 (the abscissa of the point). This, referring again to the analysis in Section 2, relates to an optimization with a slightly broader scope (more global) than that of the conventional expected improvement criterion ($w = 0.5$).

Due to the high computational expense of generating these plots, in most cases it was impractical to run the tests for every possible initial DoE size and for a very large number of different weightings – the density maps are therefore regressors through the actual data points. For each initial DoE size the optimizer has been run with 11 different weightings, covering the range between 0 and 1 with increments of 0.1. Each set of runs was stopped after some of the runs achieved an optimal (or very close to the optimum) solution – in the case of Figure 6, for example, after 11 evaluations of the objective function.

Generally the optimizer "weeds out" the very poor regions of the search space fairly rapidly. In other words, during the initial stages of the search very substantial progress is made towards the optimum, leaving

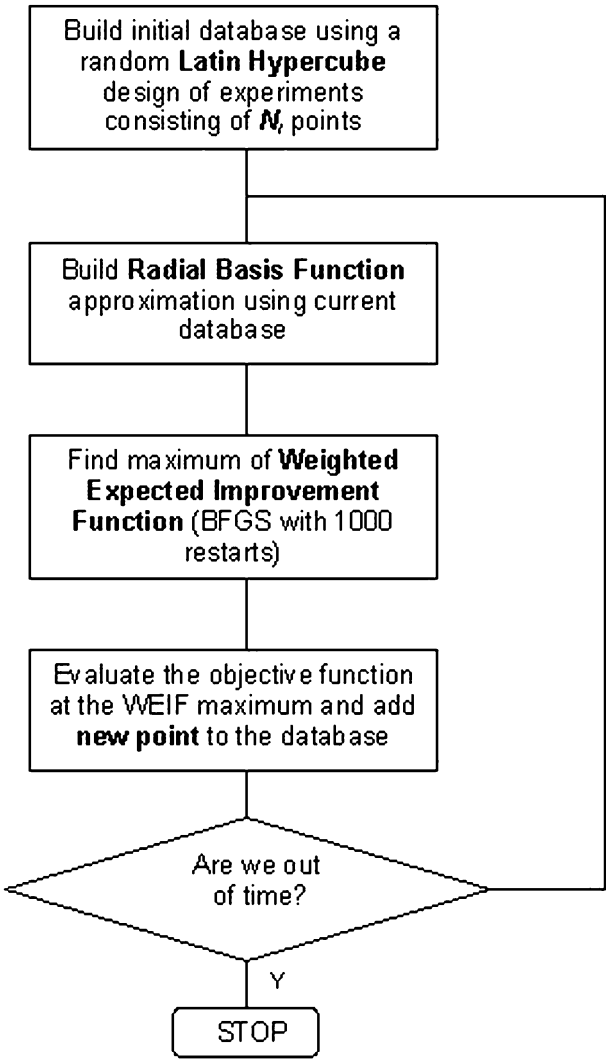

Figure 5. Optimization algorithm based on the WEIF infill sample selection criterion.

comparatively little room for variation in the crucial "fine-tuning" phase (with the exception of runs with very bad parameter choices, which make very slow progress throughout the search process). Therefore, in order to distinguish between the various areas belonging to "relatively good" initial DoE sizes and weightings, the density map is based on a logarithmic scale (as shown by the density bar adjacent to each figure).

Let us now examine Figures 6–9 in more detail, first from the perspective of the WEIF weighting. The Sphere function is the easiest of our testset. As Figure 6 illustrates, for a fairly wide range of weightings and

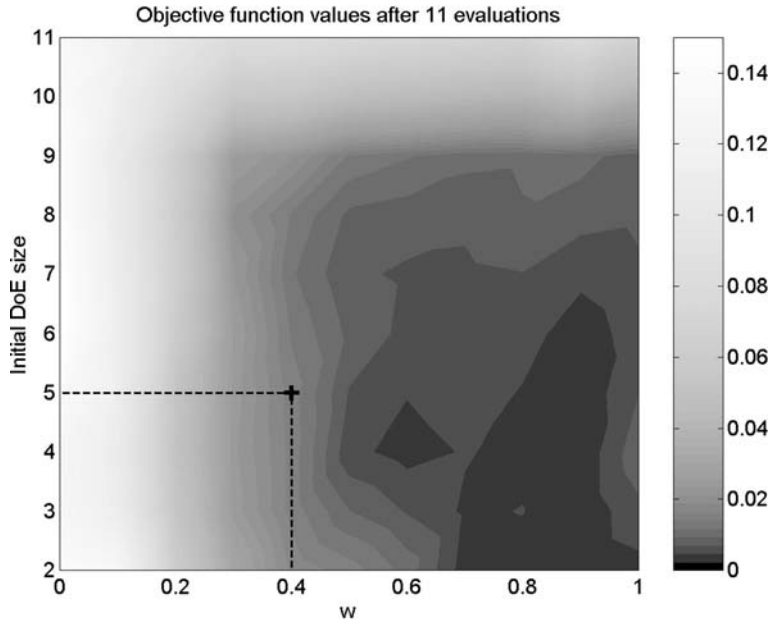

Figure 6. Log-scale colourmap of objective function values reached by the optimizer after 11 evaluations of the 5-variable Sphere function, using various WEIF weightings (horizontal axis) and initial samples of various sizes (vertical axis). The darker areas correspond to better objective values.

initial DoE sizes the problem is solved after 11 evaluations – as indicated by the large black area in the lower right-hand corner of the plot. As expected, a fairly localized search (with weightings ranging from 0.7 to 1) gives the best results.

A different landscape, the considerably multimodal Modified Rosenbrock function, yields a dramatically different plot (see Figure 7). On this occasion the black area emerging after 35 evaluations of the objective function is centered around a weighting of 0.3 for small initial sample sizes and leans slightly towards $w = 0.4$ as we move up into the zone of 15–20 point initial DoEs.

The dark region indicating good performance is even further to the left on the plot showing the objective values after 50 evaluations of the 5-variable version of the highly multimodal Ackley function (Figure 8). Clearly, the emphasis needs to shift towards exploration, when the number of potentially misleading local optima is as high as in this case. The contrast between the center of the plot and the dark area on the left (weightings ranging from 0.2 to 0.4) shows that using the normal expected improvement criterion would lead to relatively poor performance here. Further increasing the number of local optima can push the optimum weighting all the way down to zero. Evidence of this can be found on Figure 9,

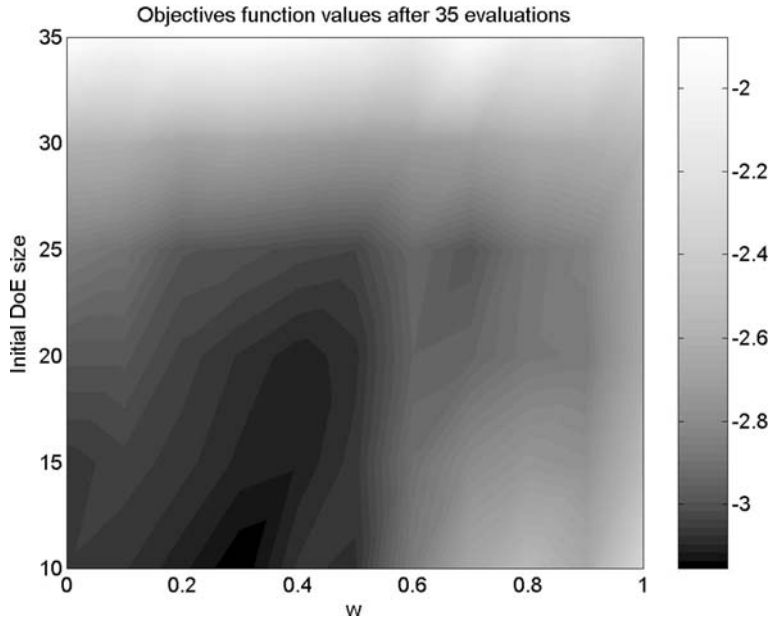

Figure 7. Log-scale colourmap of objective function values reached by the optimizer after 35 evaluations of the 5-variable Modified Rosenbrock function, using various WEIF weightings (horizontal axis) and initial samples of various sizes (vertical axis). The darker areas correspond to better objective values.

a snapshot of objective values after 60 evaluations of the same function (Ackley), this time in 10 dimensions.

We note here that this last plot is only averaged over 10 runs, due to its high computational expense. The optimizer was run for 8 different initial DoE sizes $(10, \ldots, 45)$, in each case for 11 different weightings $(0, \ldots, 1,$ increments of 0.1). With an initial DoE size of 10 the RBF model needs to be retrained and rebuilt 50 times (to reach the total objective function evaluation count of 60), starting from 15 points requires 45 constructions of the model, etc. This amounts to $50 + 45 + \cdots + 15 = 260$ runs of the training procedure for each WEIF weighting factor, that is $260 \times 11 = 2860$ across the entire range. Averaging over 10 runs thus gives a total figure of 28600 models that need to be trained. With one such procedure taking, on average, around 60 s on a PIII processor for a 10-variable problem, the required CPU time works out to 429 h for this plot alone.

The other aspect of the performance of the algorithm that we have looked at is the optimum size of the initial Latin Hypercube sample. A common conclusion that can be gleaned from all of the plots we have examined is that the algorithm becomes inefficient if the size of the initial sample exceeds about 60% of the total computational budget. This confirms our intuition, as formulated in the introductory section: if the

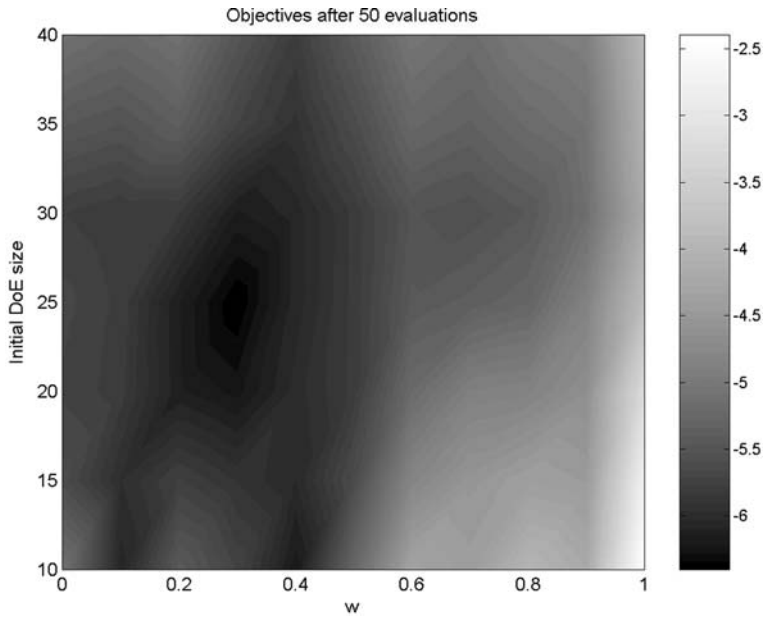

Figure 8. Log-scale colourmap of objective function values reached by the optimizer after 50 evaluations of the 5-variable Ackley function, using various WEIF weightings (horizontal axis) and initial samples of various sizes (vertical axis). The darker areas correspond to better objective values.

size of the initial sample is too large we are likely to waste points by placing them simply in a space-filling manner, instead of using information gained from the objective values of previous points (via some approximation model-based criterion). The opposite problem becomes evident when looking at either of the two Ackley plots (Figures 8 and 9). A very small initial DoE sample (10–15 points) often renders any approximation-based criterion almost entirely meaningless – as the contrast between this region and the much darker one above it (20–30 initial points) indicates, more can be gained by at least ensuring that these points fill the space uniformly, without using the objective values of the other points to decide on their location.

This phenomenon, however, only manifests itself on landscapes of very high complexity. In the majority of the cases studied here, although the small DoEs still do not contain sufficient information to allow the construction of an accurate model, the prediction based on them can offer some guidance on the choice of points, at least as valuable as choosing the points on the space-filling criterion. In summary it can be said that, based on this (admittedly limited) set of test functions, a safe choice for an initial sample size is around 35% of the available computational

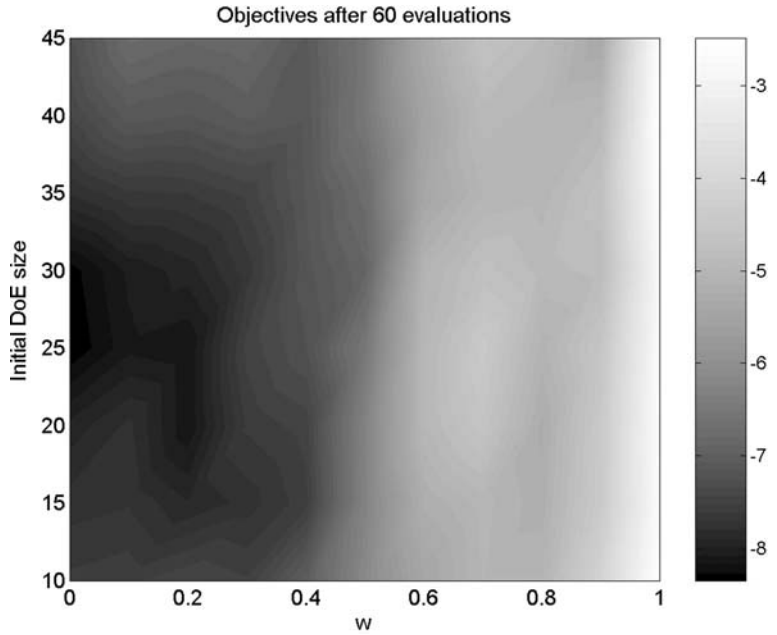

Figure 9. Log-scale colourmap of objective function values reached by the optimizer after 60 evaluations of the 10-variable Ackley function, using various WEIF weightings (horizontal axis) and initial samples of various sizes (vertical axis). The darker areas correspond to better objective values.

budget (one obvious exception to this rule is where such a choice would lead to a number of calculations that did not efficiently fill the available computing facilities – in such cases using more points would probably be more sensible).

Looking at the weighting factor in conjunction with the initial sample size, we note that the decision on their values can be made, in general, independently. With few exceptions the boundaries of the dark isles on the plots are roughly horizontal or vertical; thus for practical purposes we do not need to worry about correlations between the two factors that influence the runs. Confronted with a new practical application, it is therefore our recommendation that about 35% of the total computational budget be used on the initial Latin Hypercube sample, as this is within the black regions of all the plots considered here. This, however, only appears to be necessary for very deceptive, highly multimodal problems – for most functions it is safe (and indeed, sometimes marginally better from a performance point of view) to start from a very small DoE. We also note here that although random Latin Hypercubes were used throughout the experiments described above, in order to reduce the effect of chance on the results, in a real one-off application where the goal is to obtain the best possible optimum, one should use instead

a Latin Hypercube design optimized on some space-filling criterion (e.g., a maximum–minimum distance criterion). The choice of which weighting to use for the selection of the remaining 65% of the points ultimately comes down to the judgment and experience of the analyst. Relatively few real-life problems generate landscapes as highly multimodal as the Ackley function. Nevertheless, if this appears to be the case (based on previous experience on similar problems), one is well advised to keep the scope of the search fairly global ($w \in [0, 0.3]$). For problems where one can be reasonably confident of the accuracy of the initial prediction, values in the range [0.2, 0.5] are recommended, depending on the modality of the function. Finally, $w = 0.5$ should only be exceeded when one is confident that the landscape is of low modality. Many real-life engineering problems exhibit simple, unimodal behavior – in these cases running the WEIF-based optimizer with a weighting of around 0.9 can be expected to give good results. In the following section we discuss an application of this nature.

5.2. A "REAL-LIFE" UNIMODAL PROBLEM: GEOMETRIC OPTIMIZATION OF A SPOKED STRUCTURE

In this case study we consider the optimization of the spoked structure shown in Figure 10 (left). This model is made up entirely of beam elements whose thickness can be altered in a variety of ways. The part of the structure being optimized is shown on the right-hand side of the figure. Six design parameters define the geometry, five of which describe the ring cross section while the sixth describes the spoke sections. The rest of the model simply enforces suitable boundary conditions. Our industrial collaborators provided realistic loadings to place on the structure.

The goal here is to minimize the maximum von Mises stress (computed using the ProMecanica^{TM} package) within the structure such that the

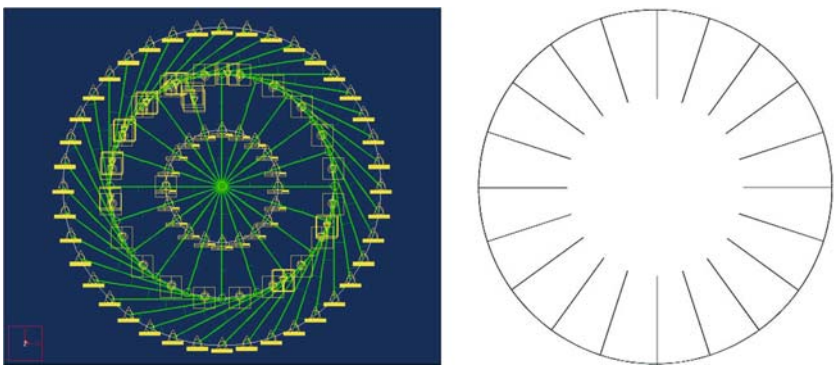

Figure 10. Full FE model (left) and part of structure to be optimized (right).

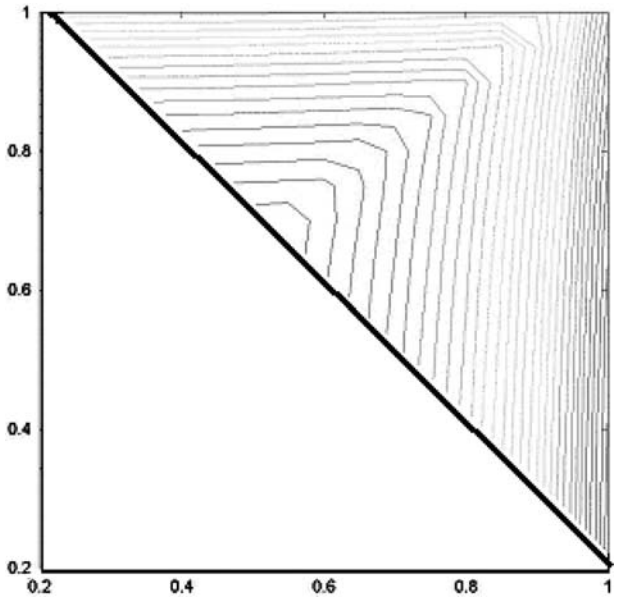

Figure 11. Two-dimensional section through the objective function of the structural testcase. The feasible region is delimited by the linear mass constraint.

weight does not exceed a predefined value. Calculating the stress involves solving the finite element problem at a cost of about 100 s of CPU time, the weight is simply the result of evaluating an exact linear model (at negligible cost) and thus can be calculated for each sampled point without incurring significant computational overheads. The WEIF updates are performed in the manner described in Section 4.

To check our conjecture that this problem is likely to generate a simple, unimodal objective we have computed a two-variable slice through the six-dimensional landscape – this is shown in Figure 11. The contour plot confirms the conjecture, as it shows a single minimum on the constraint boundary. Of course, in general we do not have the luxury of generating such plots (if we had we would not need an optimizer) – here we needed this insight to underpin our subsequent conclusions with respect to the choice of the WEIF weighting.

Figure 12 shows the optimizer performance map – a snapshot of average best objective function (stress) values after 40 evaluations of the stress and weight functions. The dark region is on the right-hand edge of the plot, indicating the need for a very localized search ($w > 0.8$). With regards to our previous conclusion about the choice of the initial sample size, we note that 35% of the total budget (14 points) is again inside the high optimizer efficiency region – thus it is a safe choice on this problem as well.

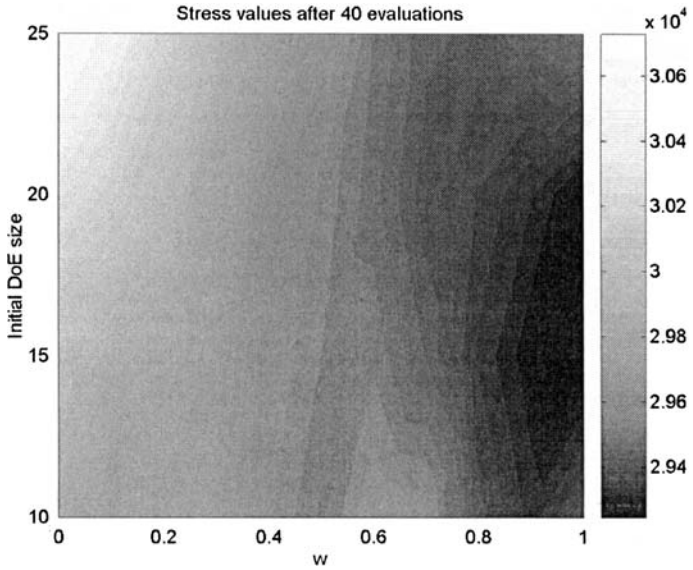

Figure 12. Log-scale colormap of objective function values reached by the optimizer after 40 evaluations of the stress function (spoke structure).

5.3. A CASE OF HIGHER MODALITY: VIBRATION OPTIMIZATION OF A TWO-DIMENSIONAL STRUCTURE

Our final study concerns optimization of the frequency response of a two-dimensional structure of the sort that may be found in girder bridges, tower cranes, satellite booms, etc. (Renton, 1999). It consists of 40 individual Euler–Bernoulli beams connected at 20 joints. Each of the 40 beams has the same properties per unit length.

Initially the boom was designed and analyzed for a regular geometry, where each beam was either 1 m or 1.414 m in length, as shown in the top section of Figure 13. The joints at points (0,0) and (0,1) are fixed, i.e., they are fully restrained in all degrees of freedom, all other joints are free to move. The structure is excited by a point transverse force applied halfway between points (0,0) and (1,0) (as indicated by the arrow on Figure 13). The vibrational energy level was found for the right-hand end vertical beam using matrix receptance methods based on the Green functions of the individual beam elements, which are set up to calculate the forces and velocities at the joints (Keane, 1995). This approach allows for a quick calculation of the energy flows around the structure. The results of the analysis have been validated experimentally (Keane and Bright, 1996).

The objective was the minimization of the frequency averaged response of the beam in the range 150–250 Hz.

For the purposes of this study the x and y locations of the two mid-span points were allowed to move during the optimization within squares

having sides of 0.5 m, centered on the initial points (see Figure 13), thus generating a four-dimensional optimization problem. The rest of the structure remained unchanged.

Again, to make the correlation between the optimum expected improvement weighting and the complexity of the landscape clearer, we have produced a two-variable slice through the design space. Figure 14 shows a contour plot of the energy level function, as measured on the right-hand end vertical beam.

As expected, this low modality (but no longer unimodal) problem dictates a weighting higher than 0.5, but lower than 0.8 (see Figure 15). In fact the dark region of the corresponding weighting-DoE size map is centered around $w = 0.7$.

As far as the optimum population size is concerned, 35% (10 points) is still a safe choice, although in this case the performance is just as good if one starts with a very small initial DoE.

6. A variable bias update strategy

We have seen in Section 5 how a rough knowledge of the complexity of the objective function can prove to be a valuable aid in choosing the right global-local bias (i.e., the weighting w) for the global optimization process. However, it is not uncommon in engineering design practice that very little is known about the nature of the objective, in which case there is no

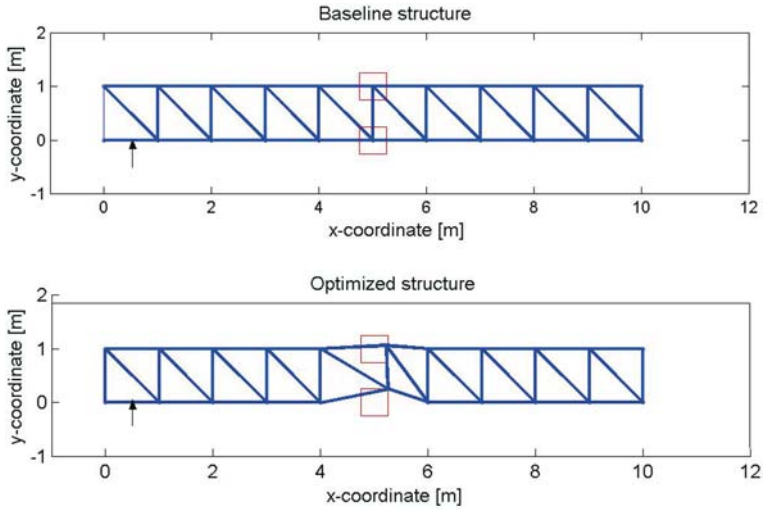

Figure 13. The two-dimensional truss in its "baseline" form (top) and in optimized form (bottom). The squares around the mid-span points indicate the ranges in which the locations of the joints were allowed to move during the optimization process.

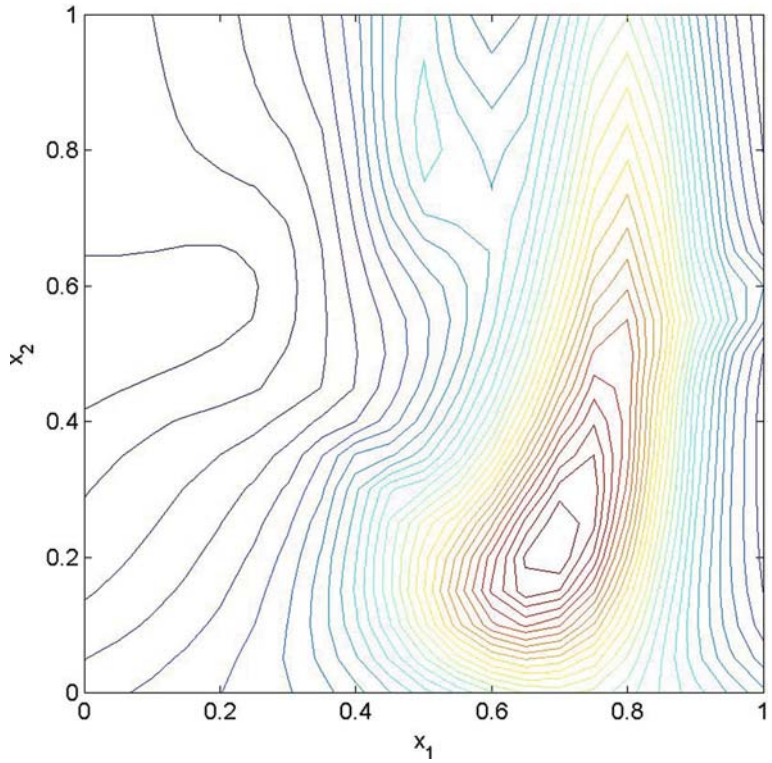

Figure 14. Two-dimensional section through the vibrational energy level function of the truss testcase. (x_1 is the x-coordinate of joint, x_2 is the y-coordinate of the same joint).

obvious alternative means of selecting the best bias without incurring significant computational expense in doing so.

For such situations Gutmann (2001) suggests cycling through the available range of global-local balances as the search progresses. He picks an infill point using a highly global setting of his criterion, the following five points to be sampled being selected by gradually shifting the balance towards exploitation. This six-step pattern is then repeated for the rest of the search. Since the same bias variation is used for all functions, the need for choosing the bias-related runtime parameter(s) is eliminated.

Here, we suggest implementing this heuristic by cycling through the pattern $w = \{0.1, 0.3, 0.5, 0.7, 0.9\}$. Thus, like Gutmann, we start from an exploratory weighting and we move towards exploitation – this pulsating search scope pattern is then repeated until we run out of time or some other stopping criterion is met.

In order to gauge the computational efficiency of this algorithm, we ran it on the Dixon–Szegö test problem set (Dixon and Szegö, 1978). The main reason for choosing this as a basis for benchmarking was the availability of an abundance of historical search performance data that we could

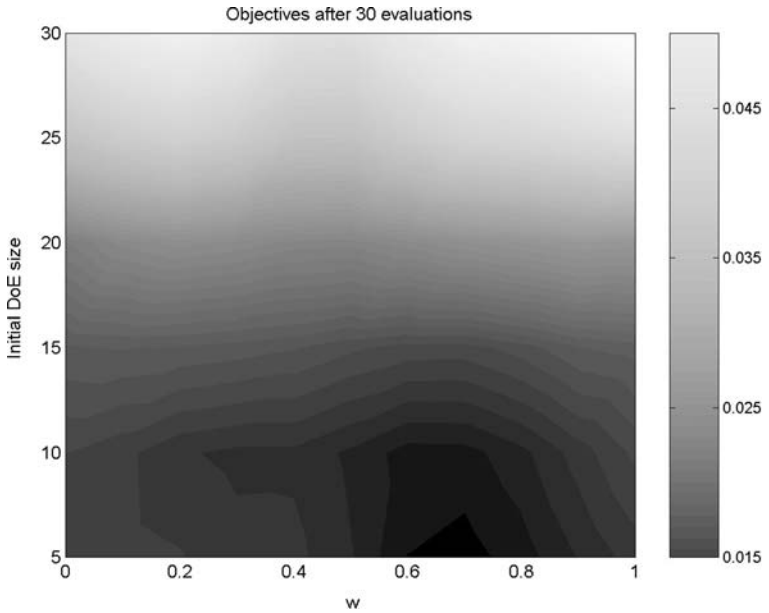

Figure 15. Log-scale colormap of objective values reached by the optimizer after 30 evaluations of the vibrational energy level function (two-dimensional truss).

Table I. Main features of the Dixon–Szegö test problems

Test function	Dimensionality	Local minima	Global minima
Branin	2	3	3
Goldstein-Price	2	4	1
Hartman 3	3	4	1
Hartman 6	6	4	1
Shekel 5	4	5	1
Shekel 7	4	7	1
Shekel 10	4	10	1

compare our performance figures with. Amongst others, Jones et al. (1998), Björkman and Holström (1999) and Gutmann (2001) run their techniques on most of the Dixon–Szegö functions – we use the same test problems, the most important features of which are summarized in Table I.

Table II contains a comparison between the convergence figures of our cyclic bias variation algorithm and those of the RBF-based cyclic search of Gutmann (2001), the DIRECT algorithm (results from Björkman and Holström (1999)) and the ubiquitous EGO technique introduced by Jones et al. (1998). Specifically, the numbers of objective function evaluations are shown for each function, which are required by the optimizers to achieve an actual relative error of 1% or better (see the caption of the table for the exact definition). Each result is averaged over 10 runs, where the runs have

Table II. Comparative optimizer performance table showing the number of evaluations required by the various optimizers to get to within 1% of the global optima of the Dixon-Szegö test functions (the error of convergence is defined as $E = 100(f_{\min} - f_{\text{global}})/|f_{\text{global}}|$, where $f_{\min}$ is the current best objective function value and f_{global} is the global optimum of the function).

	Evaluation count when $E < 1\%$			
Test function	WEIF cyclic	Gutmann	DIRECT	EGO
Branin	34	44	63	28
Goldstein-Price	32	63	101	32
Hartman 3	28	25	83	35
Hartman 6	33	112	213	121
	Best of 10 runs			
Shekel 5	43	76	103	–
Shekel 7	84	76	97	–
Shekel 10	63	51	97	–

The WEIF cyclic bias variation algorithm results are averages over 10 runs, except those for the Shekel functions, which are referring to the best of 10 runs. The runs were started from an initial set of 10 training points, arranged in a latin hypercube experimental design.

been started from different, randomly generated latin hypercube experimental designs. As before, the purpose of this experimental setup was to eliminate the effect of chance, i.e., the effect of distorted performance figures caused by one or more points of the initial design landing near the global basin.

As the table shows, the WEIF-based cyclic bias variation algorithm works well on most of the test functions that we have experimented with here. The Shekel functions might be considered to be an exception to this – here the objective values after 150 evaluations for versions 5, 7 and 10 were still short of the global optima (by 13.88%, 20.09% and 11.26%, respectively). In each case, however, there was at least one run that did reach the 1% threshold within this budget – the performance figures for the best of these are shown in the table. This indicates that, like many other approximation-based algorithms, a search based on the WEIF update scheme may be inefficient on "needle in a haystack" type problems (such as Shekel's "foxholes") – thankfully, such problems are relatively rare in engineering design optimization. We also note here that the comparative table presented here should be considered in the light of the fact that the results for Gutmann's algorithm and EGO do not take into account the variability of the performance resulting from the variability in the choice of the initial sampled points (this can make a particularly large difference when the relative area of the basins of attraction is very small, as in the case of the Shekel function family).

7. Conclusions and future work

The aim of this work has been to provide guidance on setting up optimization runs based on RBF approximation models. Central to this is the

introduction of a criterion that allows easy control of the scope of the search when selecting infill sample points. We have looked at the effects of biasing the infill selection via this criterion, either towards global exploration or towards exploitation of promising areas, through a set of empirical tests.

The two aspects we were most interested in were the selection of an appropriate weighting factor (which controls the scope of the criterion and thus the scope of the search) and the selection of the size of the initial DoE set of sampled designs. One of the most important conclusions of these experiments was the relatively high importance of these factors from the point of view of search efficiency, as highlighted by the sharp contrasts seen on some of the optimizer performance maps. However, these maps also indicate that there is some safety margin in choosing the two parameters.

Naturally, there is always room for further refinement and verification of these guidelines. The set of objective functions examined here is fairly limited and the results relevant to the choice of the initial sample size are based on a single computational budget in each case. Other types of approximation models could also be considered, including gradient-enhanced global approximations (where the objective function gradients can be obtained cheaply). Although the computational expense of building such maps can be relatively high, they are worth the effort if one is often confronted with optimization problems belonging to the same class and therefore such fine-tuning of the optimizer can be justified.

For those cases where no information is available on the complexity of the objective function, we have assessed the performance of a variable global–local bias scheme. The results obtained on a set of test functions (compared with other approximation-based techniques) are encouraging and indicate that the weighted expected improvement criterion can play a significant role even when we have no prior knowledge of the problem under scrutiny. We note here that future work in this area could include an extension of the variable bias scheme for parallel architectures, where each set of parallel updates could contain points selected with different values of the weighting (covering a range of balances from global to local).

Acknowledgements

This work has been supported by the University Technology Partnership for Design, which is a collaboration between BAE Systems, Rolls-Royce and the Universities of Cambridge, Sheffield and Southampton. The authors would like to thank Rolls-Royce for providing the application presented in Section 5.2. We are also grateful to Alexander Forrester and to an anonymous reviewer, whose suggestions proved very valuable.

References

Ackley, D.H. (1987), *A Connectionist Machine for Genetic Hillclimbing*, Kluwer Academic Publishers, Boston.

Audet, C., Dennis, J.E., Moore, D.W., Booker, A. and Frank P.D. (2000), A surrogate-model-based method for constrained optimization, In: 8th Proceedings of the *AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Long Beach, CA.

Björkman, M. and Holström, K. (1999), Global optimization using the DIRECT algorithm in Matlab, *Advanced Modeling and Optimization* 1(2), 17–28.

Dixon, L.C.W. and Szegö, G. (1978), The Global optimization problem: an introduction, In Dixon, L.C.W. and Szego, G. (EDS.), *Towards Global Optimization*, North Holland, Amsterdam, 2, pp. 1–15.

Fiacco, A.V. and McCormick, G.P. (1968), *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Wiley, New York.

Gibbs, M.N. (1997), *Bayesian Gaussian Processes for Regression and Classification*, PhD thesis, University of Cambridge.

Gutmann, H.M. (2001), A radial basis function method for global optimization, *Journal of Global Optimization* 19(3), 201–227.

Jones, D.R. (2001), A taxonomy of global optimization methods based on response surfaces, *Journal of Global Optimization* 21, 345–383.

Jones, D.R., Schonlau, M., and Welch, W.J. (1998), Efficient global optimization of expensive black-box functions, *Journal of Global Optimization* 13, 455–492.

Keane, A.J. (1995), Passive vibration control via unusual geometries: the application of genetic algorithm optimization to structural design, *Journal of Sound and Vibrations* 185(3), 441–453.

Keane, A.J. and Bright, A.P. (1996), Passive vibration control via unusual geometries: experiments on model aerospace structures, *Journal of Sound and Vibrations* 190(4), 713–719.

Mackay, M.D., Beckman, R.J. and Conover, W.J. (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, *Technometrics* 21, 239–245.

Mockus, J., Tiesis, V. and Zilinskas, A. (1978), The application of bayesian methods for seeking the extremum, *Towards Global Optimization*, North Holland, Amsterdam, 2, 117–129.

Montgomery, D. (2000), *Design and Analysis of Experiments*, 5th edn, Wiley, New York.

Renton, J.D. (1999), *Elastic Beams and Frames*, Camford Books.

Sasena, M.J., Papalambros, P. and Goovaerts, P. (2002), Exploration of metamodeling sampling criteria for constrained global optimization, *Engineering Optimization* 34, 263–278.

Schonlau, M. (1997), *Computer Experiments and Global Optimization*, PhD thesis, University of Waterloo, Canada.

Sóbester, A., Leary, S.J. and Keane, A.J. (2004), A parallel updating scheme for approximating and optimizing high fidelity computer simulations, *Structural and Multidisciplinary Optimization* 27, 371–383.

Sobol, I.M. (1979), On the systematic search in a hypercube, *SIAM Journal of Numerical Analysis* 16, 790–793.

Trosset, M.W. and Torczon V. (1997), Numerical optimization using computer experiments, technical report TR-97-38, ICASE, NASA Langley Research Center, Hampton, Virginia.

Watson, A.G. and Barnes, R.J. (1995), Infill sampling criteria to locate extremes, *Mathematical Geology* 27(5), 589–608.