
Journal of Global Optimization **21:** 345–383, 2001.

A Taxonomy of Global Optimization Methods Based on Response Surfaces

DONALD R. JONES
General Motors Corporation, Mail Code 480-305-200, 6440 East 12 Mile Road, Warren, MI 48090, USA (e-mail: don.jones@gm.com)

Abstract. This paper presents a taxonomy of existing approaches for using response surfaces for global optimization. Each method is illustrated with a simple numerical example that brings out its advantages and disadvantages. The central theme is that methods that seem quite reasonable often have non-obvious failure modes. Understanding these failure modes is essential for the development of practical algorithms that fulfill the intuitive promise of the response surface approach.

Key words: global optimization, response surface, kriging, splines

1. Introduction

Within the engineering community, response surfaces are gaining a popularity as a way of developing fast surrogates for time-consuming computer simulations. By running the simulations at a set of points (experimental design) and fitting response surfaces to the resulting input-output data, we obtain fast surrogates for the objective and constraint functions that can be used for optimization. Further time is saved by exploiting the fact that all runs used to fit the surfaces can be done in parallel, that runs can be started before one has even formulated the problem, and that runs can be reused when solving modified versions of the original problem (e.g., different constraint limits). One indication of the growing interest in this topic is that, at a recent *Symposium on Multidisciplinary Analysis and Optimization* (2000), no less than 13 papers dealt with the use of 'response surfaces' or 'surrogates' in optimization. In this paper, we will discuss the forces driving the interest in response surfaces and review the methods that have been proposed so far.

The appeal of the response-surface approach goes beyond reducing computation time. Because the approach starts with an experimental design, statistical analyses can be done to identify which input variables are the most important (highest contribution to the variance of the output) and 'main effect plots' can be created to visualize input-output relationships (see Booker, 1998; Jones et al., 1998). Using the surfaces as fast surrogates also makes it possible to quickly compute tradeoff curves between competing objectives. Multidisciplinary problems can be handled by linking response surfaces based on different engineering disciplines (e.g., crashworthiness, structures, durability). Finally, in situations where no computer model is available, response surfaces provide a way to compute 'trans-

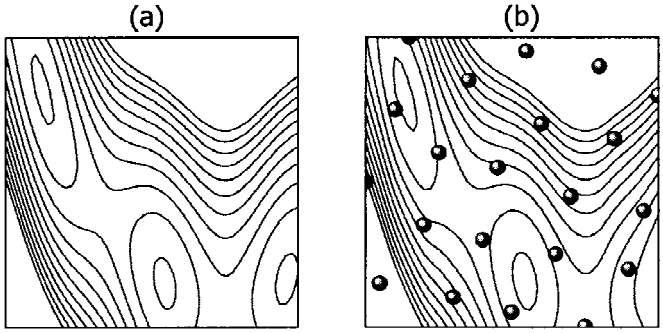

Figure 1. (a) Contours of the Branin test function. (b) Contours of a kriging surface fit to 21 points (shown as spheres).

fer functions' between inputs and outputs. These transfer functions can then be embedded in larger computer simulations or used directly for robust design and optimization.

The intuitive promise of the response surface approach is illustrated in Figure 1. On the left of the figure we show the contours of the two-dimensional Branin test function Dixon and Szego, 1978. On the right, we show the contours of a 'kriging' response surface fit to 21 points (shown as spheres). The kriging predictor is so accurate that some people do not even notice the difference between the contours in Figures 1(a) and 1(b). It is clear that we should be able to locate the global optimum quite accurately with only a few more function evaluations.

Existing approaches that use response surfaces for global optimization can be classified based on the type of response surface and the method used to select search points. In the taxonomy shown in Figure 2, seven methods are identified that will be the subject of the remaining sections in the this paper. The meaning of the entries in the taxonomy will become clearer as we proceed, but a few words now will foreshadow some of the key messages and conclusions.

At the highest level, response surfaces can be differentiated based on whether they are non-interpolating (minimize sum of squared errors from some pretermined functional form) or interpolating (pass through all points). We will show that non-interpolating surfaces, such as fitted quadratic surfaces, are unreliable because the surface may not sufficiently capture the shape of the function. It is better to use surfaces that interpolate the data with a linear combination of 'basis functions.' Among basis-function methods, one can distinguish methods in which the basis functions are fixed (thin-plate splines, Hardy multiquadrics) and those in which the basis functions have parameters that are tuned (kriging). In addition to having tuned basis functions, kriging has a statistical interpretation that allows one to construct an estimate of the potential error in the interpolator. This measure of potential error plays a key role in Methods 3, 4, and 5 (which is why those methods are not available for the non-kriging methods in Figure 2).

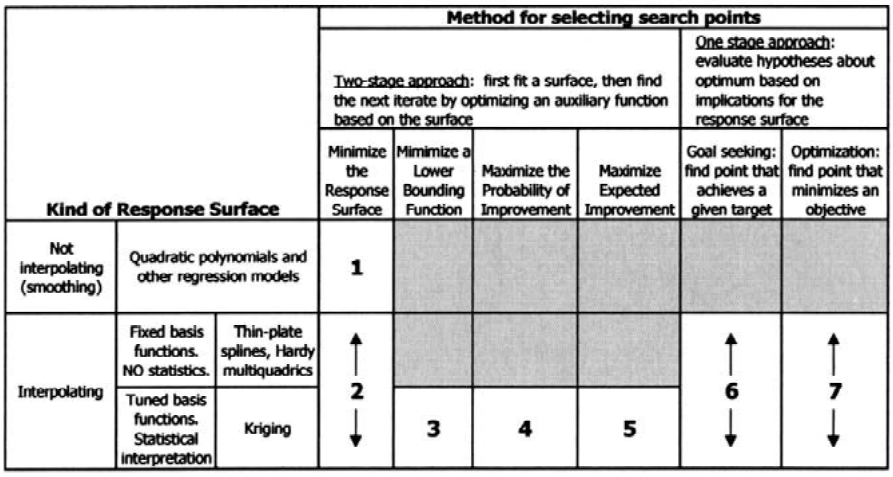

Figure 2. Taxonomy of response-surface-based global optimization methods. The seven methods shown are discussed in the text.

As for selecting search points, a key distinction will be between two-stage and one-stage methods. Most (but not all) current approaches are two-stage methods. In the first stage, one fits a response surface, estimating any required parameters. In the second stage, these parameters are treated as if they are 'true' and the surface is used to compute new search points. The potential pitfall with two-stage methods is that the initial sample may give a misleading picture of the function. As a result, one may grossly underestimate the error in the response surface and either stop prematurely or search too locally.

One-stage methods skip the initial step of fitting a surface to the observed data. Instead, the mathematical machinery of response surfaces is used to evaluate hypotheses about the location of the optimum. For example, the 'credibility' of the hypothesis that the optimum occurs at a point x^* with function value f^* may be determined by examining the properties of the best-fitting response surface that passes through the observed data *and* the point (x^*, f^*). At an intuitive level, the smoother is this surface, the more credible is the hypothesis (we will make this notion precise later). The key thing to note is that the credibility of the hypothesis is *not* based on parameters obtained by fitting a surface to the observed data alone—parameters that may be greatly in error if the initial sample is sparse and misleading.

In what follows, we will discuss all seven methods using graphical examples. As we will see, many of these methods have non-obvious failure modes. As we move from Method 1 to Method 7, we will progressively remove potential failure modes at the cost of increasing algorithmic complexity. Although the focus of this paper is on global optimization, along the way we will point out some highly successful ways in which response surfaces have been used for *local* optimization.

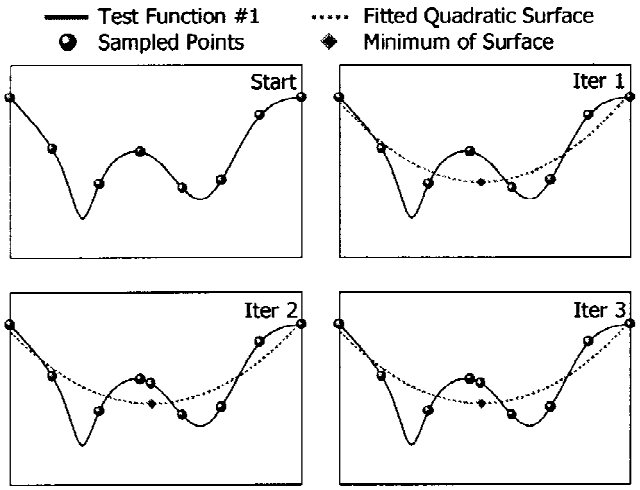

Figure 3. In Method 1, one fits a quadratic surface, finds the minimum of the surface, evaluates the function at this point, and iterates. The procedure may not even find a local minimum.

In a concluding section, we review the lessons learned and the opportunities for further work in this area.

2. Minimizing a quadratic surface

In Method 1, we begin by sampling the function according to some experimental design. In each iteration, a quadratic surface is fit to the sampled points, the minimum of the quadratic is found, and the function is evaluated at this point. The potential failure mode of this approach is illustrated in Figure 3. The panel labeled 'Start' shows the hypothetical function that we are trying to minimize—let us call it True Function #1—as well as eight points at which this function has intially been sampled (the dots). In Iteration 1 we fit a quadratic surface and find the minimum of this surface. Notice that the minimum of the quadratic does not even lie close to either of the function's two local minima. In Iteration 2 we have evaluated the function at the point that minimized the quadratic and updated the surface, but the new surface is largely unchanged from what it was in the previous iteration. The minimum of the new surface is nearly the same as before, and iterating the process further yields no improvement. This example shows that even for non-pathological functions, Method 1 can fail abysmally.

The problem with the quadratic surface is that adding additional points will not necessarily lead to a more accurate surface. In constrast, interpolating methods, such as the natural cubic splines shown in Figure 4, become more and more accurate as new points are added, eventually converging to the true function. In the next section we will examine what will happen if we replace the quadratic surface with such an interpolating response surface.

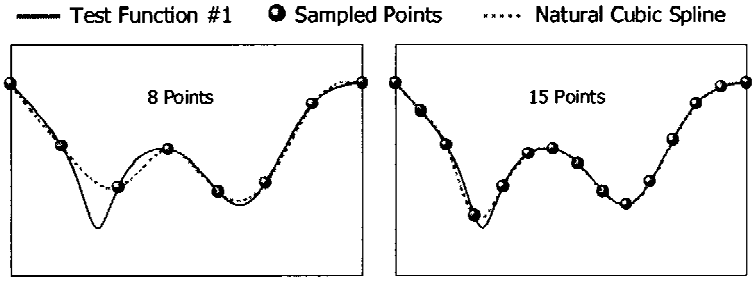

Figure 4. Data-adaptive surfaces like the natural cubic spline shown here adapt their shape to the data. As more points are added, the approximation approaches the function being fitted.

3. Minimizing an interpolating surface

Cubic splines, thin-plate spline, multiquadrics, and kriging are all methods that interpolate a set of scattered data using a linear combination of polynomial terms and special 'basis function' terms. Intuitively, the procedure is similar to expressing a complicated sound wave as a weighted sum of pure tones (sprectral decomposition). In both cases, a complicated function is expressed as a weighted sum of simple functions. Predictions can then be made by predicting the simple functions and combining these predictions using the weights.

Now let us turn to the mathematics of the basis function approach. Let us assume that we have sampled a function at n points $\mathbf{x}_i$, $i = 1, \ldots, n$, where each $\mathbf{x}_i$ is a d-dimensional vector $\mathbf{x}_i = (x_{i1}\ x_{i2}\ \cdots\ x_{id})'$. Denote the function values at these points by $y_i = y(\mathbf{x}_i)$. Also, let $\{\pi_k(\mathbf{x}) \mid k = 1, \ldots, m\}$ be a basis of the set of all polynomials in $\mathbf{x}$ of degree G. Note that $\mathbf{x}$ is multidimensional, so that each $\pi_k(\mathbf{x})$ is a weighted sum of terms like $x_1^{g_1} x_2^{g_2} \cdots x_d^{g_d}$ where $g_1 + g_2 + \cdots + g_d \leqslant G$. In the case $G = 1$, for example, a possible basis would be the set of $d+1$ functions consisting of the constant function $\pi_1(\mathbf{x}) = 1$ and the linear terms $\pi_{1+\ell}(\mathbf{x}) = x_\ell$, $\ell = 1, \ldots, d$.

Now, if we fit a surface by least squares using only the m polynomial terms, we would be back to curve fitting as in Method 1. The basis function approach differs from fitting polynomials because the interpolator not only includes these polynomial terms, but also includes n 'basis function' terms, each centered around one of the sampled points. That is, our predictor at a new point $\mathbf{x}^*$ is of the form

$$\widehat{y}(\mathbf{x}^*) = \sum_{k=1}^{m} a_k \pi_k(\mathbf{x}^*) + \sum_{j=1}^{n} b_j \varphi(\mathbf{x}^* - \mathbf{x}_j). \tag{1}$$

Possible choices for the basis function $\varphi(\mathbf{z})$ are:

$$\begin{array}{ll}
\varphi(\mathbf{z}) = \|\mathbf{z}\| & \text{(linear)} \\
\varphi(\mathbf{z}) = \|\mathbf{z}\|^3 & \text{(cubic)} \\
\varphi(\mathbf{z}) = \|\mathbf{z}\|^2 \log(\|\mathbf{z}\|) & \text{(thin plate spline)} \\
\varphi(\mathbf{z}) = \sqrt{\|\mathbf{z}\|^2 + \gamma^2} & \text{(multiquadric)} \\
\varphi(\mathbf{z}) = \exp\left(-\sum_{\ell=1}^{d} \theta_\ell \ |z_\ell|^{p_\ell}\right) & \text{(kriging)}
\end{array} \tag{2}$$

Here $\|\mathbf{z}\|$ is the Euclidean norm and, in the multiquadric case, γ is a prescribed positive constant. The basis function shown for kriging is only one possibility, but it is a popular choice that appeared in an influential article by Sacks *et al.* (1989). In this basis function, the parameters θ_ℓ and p_ℓ are assumed to satisfy $\theta_\ell \geqslant 0$ and $0 < p_\ell \leqslant 2$.

Now, to interpolate the data, we require

$$y_i = \sum_{k=1}^{m} a_k \pi_k(\mathbf{x}_i) + \sum_{j=1}^{n} b_j \varphi\left(\mathbf{x}_i - \mathbf{x}_j\right), \quad \text{for } i = 1, .., n \ . \tag{3}$$

However, these are only n equations, and we have $n + m$ unknowns (the a_k's and b_j's). To complete the system, we add the following additional m constraints:

$$\sum_{j=1}^{n} b_j \pi_k\left(\mathbf{x}_j\right) = 0 \quad \text{for } k = 1, .., m \ . \tag{4}$$

While it is clear that we need m constraints to complete the system, it is far from obvious why it is *these particular m constraints* that should be added. However, one can show that adding these particular constraints gives the interpolator a very desirable property. In particular, if the function being approximated actually *is* a polynomial of degree G or less, then the interpolator will be exact (that is, it will *be* the polynomial). More generally, these additional constraints cause the term $\sum_{k=1}^{m} a_k \pi_k(\mathbf{x}^*)$ to be a projection of the function onto the space of polynomials of degree G, and the term $\sum_{j=1}^{n} b_j \varphi\left(\mathbf{x}^* - \mathbf{x}_j\right)$ to be an interpolator of the residual, non-polynomial part of the function.

Of course, it would be natural for the reader to wonder why we should even bother to add the polynomial terms in the first place. After all, wouldn't it be simpler to simply use the n basis functions and identify the coefficients using the n equations in (3)? While this would certainly be simpler, in some cases the polynomial terms are necessary to guarantee that the system of equations is nonsingular. In particular, we need $G \geqslant 0$ in the linear and multiquadric cases and $G \geqslant 1$ in the cubic and thin-plate spline cases. The basis function for kriging has the property that the system is nonsingular even if no polynomial terms are used, though it is common to include a constant term ($G = 0$).

Some of these interpolators have properties that make them intuitively appealing. For example, the cubic spline interpolator in one dimension is the smoothest

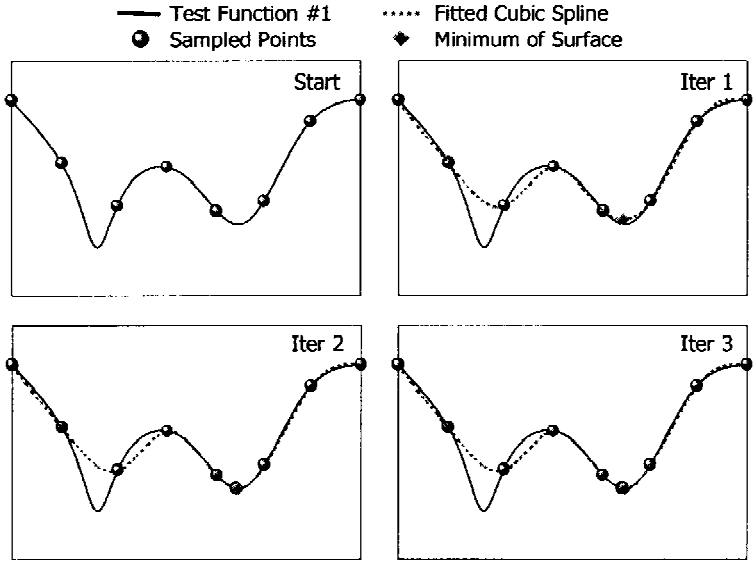

Figure 5. Illustration of Method 2 on a simple one-dimensional test function. In each iteration, a spline or kriging surface is fit to the data, the surface is minimized, and the function is evaluated at the surface minimum.

possible interpolator, in the sense that it minimizes $\int \left[\widehat{y}''(x)\right]^2 dx$ among all functions that satisfy the interpolation conditions $\widehat{y}(\mathbf{x}_i) = y_i$ and for which the integral exists and is finite. A similar smoothness property applies to the thin-plate splines in two dimensions. Kriging also has an intuitive, statistically-oriented, motivation, which we discuss later. For now, let us turn to Figure 5 and see what happens if we replace the quadratic surface in the previous example with an interpolating surface, here a cubic spline. This is 'Method 2' in the taxonomy of Figure 2.

In Iteration 1, the fitted spline is minimized at a point very near the rightmost local minimum. Further iterations improve the spline approximation in this region, so that by Iteration 3 we have found the this local minimum to great accuracy. Unfortunately, however, it is only a local minimum. The technique has missed the global minimum to the left.

At this point, many readers may say to themselves, 'OK, so we missed the global optimum. But what a wonderful and efficient method for finding a local minimum!' However, we can be easily deceived: Figure 6 shows another possible function—True Function #2—for which the spline approach would fail. In this case, the spline is minimized at a sampled point. As a result, adding the minimum of the spline to the data and updating the surface will have no effect. The method has converged, but it has converged to a point that is not even a local optimum! Of course, in practice, it is highly unlikely that the surface will be minimized *exactly* at a sampled point. Instead, it will usually be minimized a little to the left or right and, when we sample this point, we will 'see' that the function is not flat. The non-

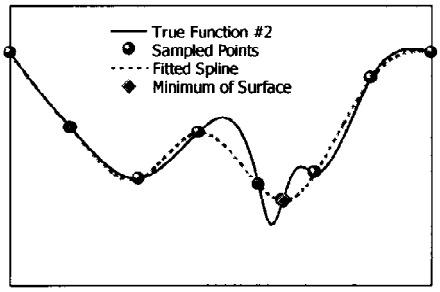

Figure 6. Method 2 can fail because a local minimum of the *surface* may not be a critical point of the *function*.

zero slope at this point will be reflected in the updated response surface, and the procedure would still make progress. On the other hand, any practical algorithm would need some criterion for stopping. If this criterion were 'stop when the minimum of the surface is within a distance $\varepsilon>0$ of a sampled point,' then the situation in Figure 6 might actually occur in practice. In any case, what Figure 6 certainly shows is that convergence cannot be guaranteed.

The previous discussion suggests that, whenever Method 2 *seems* to have converged (i.e., the minimum of surface is near a sampled point), we should nevertheless sample in a small neighborhood of the tentative solution to force the gradient of the surface to agree with the gradient of the true function (kriging can also be directly adapted to utilize derivative information; see Koehler and Owen, 1996). This idea is explored in Figure 7. We begin with the situation just described, where the minimum of the spline occurs at a sampled point. Since this means we have tentatively converged, we continue by sampling a little to the left and right of this point, since this will cause the gradient of the surface to match that of the function Having done this, further iterations quickly converge to the local minimum.

Adding 'gradient matching' whenever the minimum of the surface is near a sampled point certainly makes Method 2 more robust. Whether or not one can prove that this method, if iterated infinitely, must converge to a local optimum is an open research question.

As we have suggested, the choice of stopping rule will be important in any practical implementation of Method 2. A tempting choice would be to stop when the improvement from one iteration to the next is small, but there is no reason to believe that the sequence of sampled function values will be decreasing. For example, Figure 8 shows what happens when we use the gradient-matching technique on another possible function (True Function #3). In this case, after the gradient-matching step, we overshoot the local minimum by so much that the new point yields no improvement. It is fairly clear, however, that if we add this point to the sample and continue iterating, we will converge to the local minimum. So lack of improvement from one iteration to the next is not a reliable stopping rule.

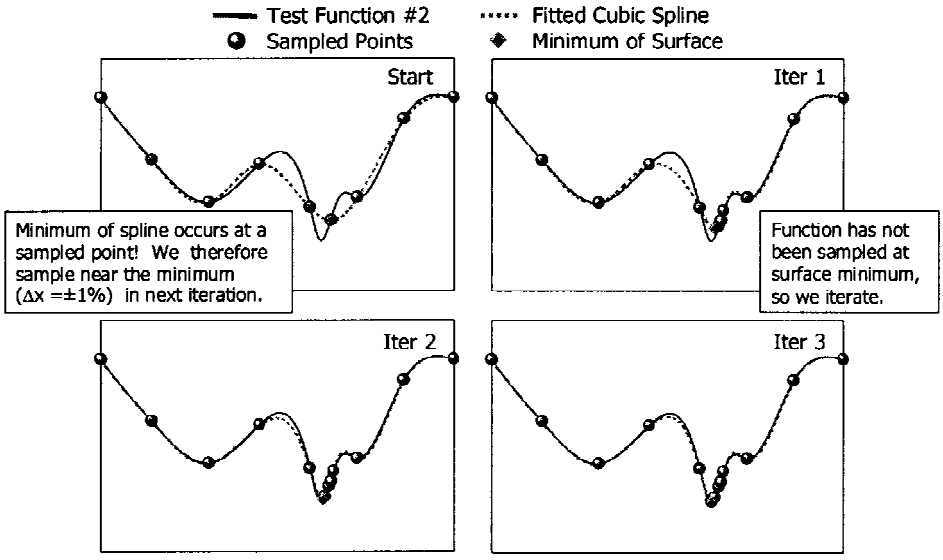

Figure 7. To prevent premature convergence in Method 2, we can force the surface gradient to agree with the function gradient whenever we tentatively converge.

Local convergence can be assured, however, if one combines gradient matching with a trust region approach. In particular, Alexandrov *et al.* (2000) develop an algorithm that uses a 'correction factor' approach to force gradient matching between the surface and function at the current iterate. To get the next iterate, the surface is optimized within a trust region around the incumbant solution. The optimum point is then sampled and, if the objective function fails to decrease, the trust region is contracted. They prove that this approach must converge to a critical point of the function.

It is perhaps best to think of Alexandrov's approach as a way of using response surfaces to *accelerate* an existing, provably convergent local optimization method—as opposed to thinking of it as a new, response-surface-based method. Alexandrov starts with the well-known trust-region algorithm; however, instead of finding the next iterate by minimizing a second-order Taylor-series approximation within the trust region, she minimizes the *response surface* within the trust region. Because the response surface will usually be more accurate than the Taylor approximation, one will usually be able to take longer steps; hence, the algorithm will proceed faster. Gradient matching is necessary to preserve the convergence properties of the trust region approach.

Response surfaces have also been used to accelerate derivative-free methods for local optimization. For example, Booker *et al.* (1999) use kriging response surfaces to accelerate the General Pattern Search algorithm of Dennis and Torczon (1991). In the original, un-accelerated version of the Pattern Search algorithm, one searches over a lattice of points around the current iterate until either one finds an

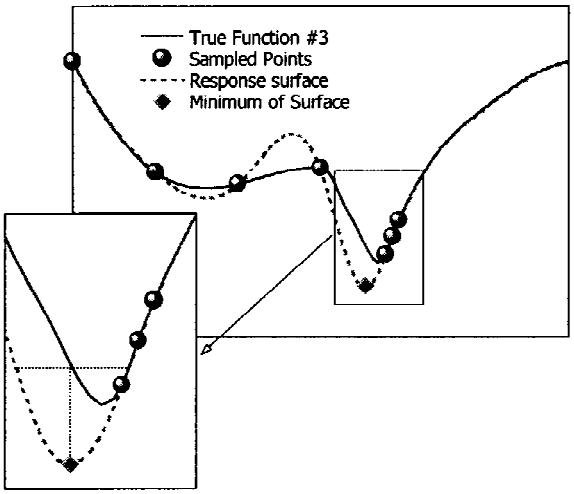

Figure 8. Even if force gradient matching between the surface and function, the minimum of the surface may not be an improving point. To remedy this problem, optimization of the surface can be restricted to a trust region around the current iterate.

improving point or until all the lattice points have been sampled. In the accelerated version, a kriging response surface is used to predict the function's value at the points in the lattice, and the points are sampled *starting with the ones that the kriging surface predicts will have the best values*. In this way, an improving point is usually found much sooner than would be the case if the lattice points were sampled in a random order. Booker *et al.* found that the reduction in function evaluations can be substantial.

While these local methods are exciting developments, we should bear in mind that the methods can only guarantee convergence to a critical point. It is entirely possible that this point will merely be a saddle point or 'flat spot' of the function. This is illustrated in Figure 9 for yet another possible true function (True Function #4). In this case, the response surface is minimized at a sampled point *and* the gradient of the function is zero at this point. Thus, all of the previous methods would have declared success. Yet we are not even at a local minimum, far less a global one.

A well known theorem by Torn and Zilinskas (1987) tells us that, in order to converge to the global optimum for a general continuous function, the sequence of iterates must be dense. Of course, this is a rather trivial method of convergence; in essence, it says that, in order to guarantee convergence to the global optimum, we must converge to every point in the domain. The practical lesson of the theorem is that any globally convergent method must have a feature that forces it to pay attention to parts of the space that have been relatively unexplored and, from time to time, to go back and sample in these regions. Methods 1–2 failed to find a global minimum in our examples because they have no such feature.

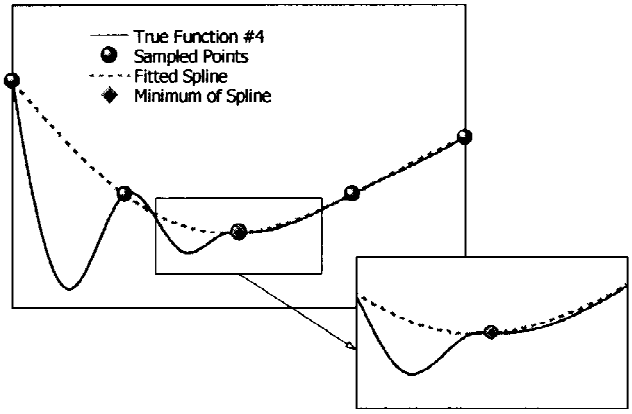

Figure 9. In this example, surface minimum is at a sampled point and the gradient of the surface and function agree. We have converged, but only to a saddle point.

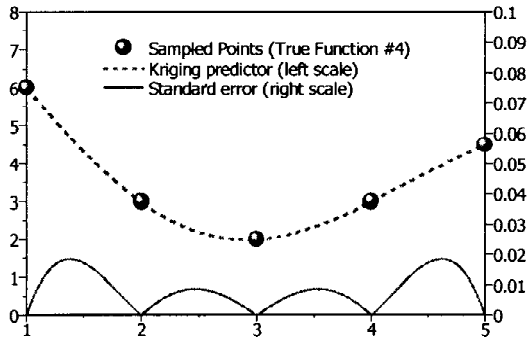

Figure 10. The kriging predictor and its standard error for True Function #4.

As mentioned in the introduction, kriging stands out from other basis function methods because it has a statistical interpretation. This interpretation not only allows us to compute an interpolator (or 'predictor'), but also allows us to compute a measure of the possible error in the predictor. Figure 10 illustrates this idea by showing a kriging surface fit to our deceptive True Function #4 of the previous example. In addition to the kriging predictor, we also show a curve labeled 'standard error.' The standard error (right hand scale) goes to zero at the sampled points, indicating that we have no uncertainty about these values. In between the sampled points, the standard error rises. Intuitively, the further we are from the nearest sampled point, the more uncertain we are about the function, and the higher is the standard error.

With kriging, we can develop search methods that put some emphasis on sampling where the standard error is high. In this way, we obtain the desired feature of 'paying attention to parts of the space that have been relatively unexplored.' Methods 3, 4, and 5 (discussed later) involve different ways of implementing this

idea. Since kriging will play a large role in this discussion, we will first digress to discuss it in more detail.

4. A gentle introduction to kriging

Most articles and textbooks describe kriging as a way of 'modeling the function as a realization of a stochastic process'. The kriging predictor is then shown to be the predictor that minimizes the expected squared prediction error subject to: (i), being unbiased and, (ii), being a linear function of the observed y_i's. (Although the predictor is *linear* in the y_i's, the coefficients can be *nonlinear* functions of $\mathbf{x}$, and hence the predictor can be nonlinear). While this is approach is rigorous and facilitates the derivation of key formulas, for most people the approach is not intuitive. In what follows, we will derive the kriging formulas using a somewhat different approach. Readers intesterested in the standard derivation may consult the article by Sacks *et al.* (1985).

Suppose we want to make a prediction at some point $\mathbf{x}$ in the domain. Before we have sampled any points, we will be uncertain about the value of the function at a this point. Let us model this uncertainty by saying that the value of the function at $\mathbf{x}$ is like the realization of a random variable $Y(\mathbf{x})$ that is normally distributed with mean μ and variance σ^2. Intuitively, this means that we are saying that the function has a typical value of μ and can be expected to vary in some range like $[\mu - 3\sigma, \mu + 3\sigma]$. Now consider two points $\mathbf{x}_i$ and $\mathbf{x}_j$. Again, before we have sampled these points, we are uncertain about the associated function values. However, assuming the function being modeled is continuous, the function values $y(\mathbf{x}_i)$ and $y(\mathbf{x}_j)$ will tend to be close if the distance $\|\mathbf{x}_i - \mathbf{x}_j\|$ is small. We can model this statistically by saying that the random variables $Y(\mathbf{x}_i)$ and $Y(\mathbf{x}_j)$ will be highly correlated if $\|\mathbf{x}_i - \mathbf{x}_j\|$ is small. In particular, we will assume that the correlation between the random variables is given by

$$\mathrm{Corr}\left[Y(\mathbf{x}_i), Y(\mathbf{x}_j)\right] = \exp\left(-\sum_{\ell=1}^{d} \theta_\ell \left|\mathbf{x}_{i\ell} - \mathbf{x}_{j\ell}\right|^{p_\ell}\right). \tag{5}$$

This correlation function has the intuitive property that if $\mathbf{x}_i = \mathbf{x}_j$, then the correlation is 1. Similarly, as $\|\mathbf{x}_i - \mathbf{x}_j\| \to \infty$, the correlation tends to zero. The θ_ℓ parameter determines how fast the correlation 'drops off' as one moves in the ℓ^{th} coordinate direction. Large values of θ_ℓ serve to model functions that are highly active in the ℓ^{th} variable; for such variables, the function's value can change rapidly even over small distances. The p_ℓ determines the smoothness of the function in the ℓ^{th} coordinate direction. Values of p_ℓ near 2 help model smooth functions, while values of p_ℓ near 0 help model rough, nondifferentiable functions.

Putting it all together, we can represent our uncertainty about the function's values at the n points using the the random vector

$$\mathbf{Y} = \begin{pmatrix} Y(\mathbf{x}_1) \\ \vdots \\ Y(\mathbf{x}_n) \end{pmatrix}. \tag{6}$$

This random vector has mean equal to $\mathbf{1}\mu$, where $\mathbf{1}$ is a $n \times 1$ vector of ones, and covariance matrix equal to

$$\text{Cov}(\mathbf{Y}) = \sigma^2 \mathbf{R}, \tag{7}$$

where $\mathbf{R}$ is a $n \times n$ matrix with (i, j) element given by Eq. (5). The distribution of $\mathbf{Y}$—which depends upon the parameters μ, σ^2, θ_ℓ and p_ℓ ($\ell = 1, .., d$)—characterizes how we expect the function to vary as one moves in different coordinate directions.

To estimate the values of μ, σ^2, θ_ℓ and p_ℓ ($\ell = 1, .., d$), we choose these parameters to maximize the likelihood of the observed data. Let the vector of observed function values be denoted

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}. \tag{8}$$

With this notation, the likelihood function may then be written as

$$\frac{1}{(2\pi)^{\frac{n}{2}} \left(\sigma^2\right)^{\frac{n}{2}} |\mathbf{R}|^{\frac{1}{2}}} \exp\left[\frac{-(\mathbf{y} - \mathbf{1}\mu)' \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\mu)}{2\sigma^2}\right]. \tag{9}$$

Choosing the parameters to maximize the likelihood function intuitively means that we want our model of the function's typical behavior to be most consistent with the data we have seen.

In practice it is more convenient to choose the parameters to maximize the log of the likelihood function, which—ignoring constant terms—is:

$$-\frac{n}{2} \log\left(\sigma^2\right) - \frac{1}{2} \log\left(|\mathbf{R}|\right) - \frac{(\mathbf{y} - \mathbf{1}\mu)' \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\mu)}{2\sigma^2}. \tag{10}$$

Setting the derivatives this expression with respect to σ^2 and μ to zero and solving, we can express the optimal values of σ^2 and μ as functions of $\mathbf{R}$:

$$\widehat{\mu} = \frac{\mathbf{1}' \mathbf{R}^{-1} \mathbf{y}}{\mathbf{1}' \mathbf{R}^{-1} \mathbf{1}} \tag{11}$$

$$\widehat{\sigma}^2 = \frac{(\mathbf{y} - \mathbf{1}\widehat{\mu})' \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\widehat{\mu})}{n}. \tag{12}$$

Substituting Eqs. (11) and (12) into Eq. (10) we get the so-called 'concentrated log-likelihood' function. Ignoring constant terms, the concentrated log-likelihood function is:

$$-\frac{n}{2}\log\left(\widehat{\sigma}^2\right) - \frac{1}{2}\log\left(|\mathbf{R}|\right). \tag{13}$$

The concentrated log-likelihood function depends only on $\mathbf{R}$ and, hence, on the correlation parameters (θ's and p's). In practice, this is the function that we maximize to get estimates $\widehat{\theta}_\ell$ and $\widehat{p}_\ell$ ($\ell = 1, .., d$). Given these estimates, we then use Eqs. (11) and (12) to compute the estimates $\widehat{\mu}$ and $\widehat{\sigma}^2$.

To understand how we can make predictions at some new point $\mathbf{x}^*$, suppose y^* were some guessed function value. To evaluate the quality of this guess, we may add the point $(\mathbf{x}^*, y^*)$ to the data as the $(n+1)^{th}$ observation and compute the 'augmented' likelihood function using parameter values obtained in the maximum likelihood estimation. As we have seen, these estimated parameters reflect the typical pattern of variation in the observed data. With these parameters fixed, the augmented log-likelihood is simply a function of y^* and reflects how *consistent* the point $(\mathbf{x}^*, y^*)$ is with the observed pattern of variation. An intuitive predictor is therefore the value of y^* that maximizes this augmented likelihood function. It turns out that this value of y^* is the kriging predictor, and we will now proceed to derive it.

Let $\widetilde{\mathbf{y}} = (\mathbf{y}' \quad y^*)'$ denote the vector of function values when augmented by the new, $(n+1)^{th}$ pseudo-observation $(\mathbf{x}^*, y^*)$. Also, let $\mathbf{r}$ denote the vector of correlations of $Y(\mathbf{x}^*)$ with $Y(\mathbf{x}_i)$, for $i = 1, \ldots, n$:

$$\mathbf{r} = \begin{pmatrix} \text{Corr}\left[Y(\mathbf{x}^*), Y(\mathbf{x}_1)\right] \\ \vdots \\ \text{Corr}\left[Y(\mathbf{x}^*), Y(\mathbf{x}_n)\right] \end{pmatrix} \tag{14}$$

The correlation matrix for the augmented data set, denoted $\widetilde{\mathbf{R}}$, is:

$$\widetilde{\mathbf{R}} = \begin{pmatrix} \mathbf{R} & \mathbf{r} \\ \mathbf{r}' & 1 \end{pmatrix}. \tag{15}$$

Now if the reader looks again at the formula for the log-likelihood function in Eq. (10), it will be clear that the only part of the *augmented* log-likelihood function that depends upon y^* is

$$-\frac{(\widetilde{\mathbf{y}} - \mathbf{1}\widehat{\mu})'\,\widetilde{\mathbf{R}}^{-1}\,(\widetilde{\mathbf{y}} - \mathbf{1}\widehat{\mu})}{2\widehat{\sigma}^2}.$$

Substituting in the expressions for $\widetilde{\mathbf{y}}$ and $\widetilde{\mathbf{R}}$, we may write this as

$$\frac{-\begin{pmatrix} \mathbf{y} - \mathbf{1}\widehat{\mu} \\ y^* - \widehat{\mu} \end{pmatrix}' \begin{pmatrix} \mathbf{R} & \mathbf{r} \\ \mathbf{r}' & 1 \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{y} - \mathbf{1}\widehat{\mu} \\ y^* - \widehat{\mu} \end{pmatrix}}{2\widehat{\sigma}^2}. \tag{16}$$

Now, from the partitioned inverse formula (Theil, 1971, p. 18), we have the following explicit expression for $\widetilde{\mathbf{R}}^{-1}$:

$$\left(\begin{array}{c|c} \mathbf{R}^{-1}+\mathbf{R}^{-1}\mathbf{r}\left(1-\mathbf{r}'\mathbf{R}^{-1}\mathbf{r}\right)^{-1}\mathbf{r}'\mathbf{R}^{-1} & -\mathbf{R}^{-1}\mathbf{r}\left(1-\mathbf{r}'\mathbf{R}^{-1}\mathbf{r}\right)^{-1} \\ \hline -\left(1-\mathbf{r}'\mathbf{R}^{-1}\mathbf{r}\right)^{-1}\mathbf{r}'\mathbf{R}^{-1} & \left(1-\mathbf{r}'\mathbf{R}^{-1}\mathbf{r}\right)^{-1} \end{array}\right).$$

Substituting this into Eq. (16), we may write the augmented log-likelihood as:

$$\left[\frac{-1}{2\widehat{\sigma}^2\left(1-\mathbf{r}'\mathbf{R}^{-1}\mathbf{r}\right)}\right](y^*-\widehat{\mu})^2+\left[\frac{\mathbf{r}'\mathbf{R}^{-1}\left(\mathbf{y}-\mathbf{1}\widehat{\mu}\right)}{\widehat{\sigma}^2\left(1-\mathbf{r}'\mathbf{R}^{-1}\mathbf{r}\right)}\right](y^*-\widehat{\mu})+\begin{array}{l}\text{terms}\\ \text{with-}\\ \text{out}y^*\end{array}. \quad (17)$$

Thus we see that the augmented likelihood is actually a quadratic function of y^*. The value of y^* that maximizes the augmented likelihood is found by taking the derivative of the above expression and setting it equal to zero:

$$\left[\frac{-1}{\widehat{\sigma}^2\left(1-\mathbf{r}'\mathbf{R}^{-1}\mathbf{r}\right)}\right](y^*-\widehat{\mu})+\left[\frac{\mathbf{r}'\mathbf{R}^{-1}\left(\mathbf{y}-\mathbf{1}\widehat{\mu}\right)}{\widehat{\sigma}^2\left(1-\mathbf{r}'\mathbf{R}^{-1}\mathbf{r}\right)}\right]=0. \quad (18)$$

Solving for y^* then gives us the standard formula for the kriging predictor:

$$\widehat{y}\left(\mathbf{x}^*\right)=\widehat{\mu}+\mathbf{r}'\mathbf{R}^{-1}\left(\mathbf{y}-\mathbf{1}\widehat{\mu}\right). \quad (19)$$

Now if we let let $\varphi(\mathbf{z})$ be the kriging basis function listed earlier in Eq. (2), then the i^{th} element of $\mathbf{r}$ is just $\varphi(\mathbf{x}^*-\mathbf{x}_i)$. If we further let $a=\widehat{\mu}$ and let b_i be the i^{th} element of $\mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\widehat{\mu})$, then the kriging predictor can be written as

$$\widehat{y}\left(\mathbf{x}^*\right)=a+\sum_{i=1}^{n}b_i\varphi\left(\mathbf{x}^*-\mathbf{x}_i\right). \quad (20)$$

Thus, the kriging predictor is indeed a linear combination of basis functions and polynomial terms (here just a constant).

Now it makes intuitive sense that we should be more confident in our predictor if the augmented log-likelihood drops off dramatically as one moves away from the optimal value of y^*. Intuitively, this means that values of y^* that are different from the predictor are much less consistent with the pattern of variation in the observed data. This is illustrated in Figure 11 which shows two possible ways in which the augmented likelihood could depend upon y^*.

In Figure 11(a), the augmented log-likelihood is fairly flat. While there is a value of y^* that maximizes this function, we can hardly be confident in this estimate because other, quite different values of y^* perform almost as well. On the other hand, in Figure 11(b) the augmented log-likelihood is strongly peaked. In this case, values of y^* that differ substantially from the predictor are much less consistent with the data, and so we can be more confident in the kriging predictor. This line of thought suggests that our estimate of the potential error in the predictor should

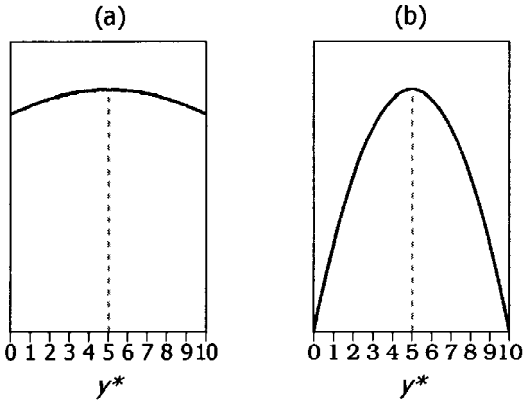

Figure 11. Two possible ways in which the augmented log-likelihood might depend upon the guessed value y^* for $y\left(\mathbf{x}^*\right)$. In both cases the best prediction is $y^* = 5$, but we would be more confident in case (b) than in case (a).

be inversely related to the curvature of the augmented log-likelihood function. Low curvature (flat function) suggests high potential error; likewise, high curvature (strongly peaked function) suggests low potential error. The curvature, in turn, can be measured by the absolute value of the second derivative of the augmented log-likelihood function with respect to y^*. From Eq. (17), we find that the absolute value of the second derivative is:

$$\frac{1}{\widehat{\sigma}^2\left(1-\mathbf{r}'\mathbf{R}^{-1}\mathbf{r}\right)}\,. \tag{21}$$

Since we have argued that the error in the predictor is inversely related to this curvature, a natural measure of potential error is the reciprocal of the curvature:

$$\widehat{\sigma}^2\left(1-\mathbf{r}'\mathbf{R}^{-1}\mathbf{r}\right)\,. \tag{22}$$

This is, in fact, very close to the formula for the mean-squared error of the predictor derived using the standard stochastic-process approach. This standard formula, which we will denote by $s^2\left(\mathbf{x}^*\right)$, is

$$s^2\left(\mathbf{x}^*\right) = \widehat{\sigma}^2\left[1-\mathbf{r}'\mathbf{R}^{-1}\mathbf{r}+\frac{\left(1-\mathbf{r}'\mathbf{R}^{-1}\mathbf{r}\right)^2}{\mathbf{1}'\mathbf{R}^{-1}\mathbf{1}}\right]. \tag{23}$$

The extra term on the right hand side of Eq. (23) can be interpreted as representing the uncertainty that stems from our not knowing μ exactly, but rather having to estimate it from the data.

The formula for $s^2\left(\mathbf{x}^*\right)$ has the intuitive property that it is zero at any sampled point. This is as it should be since we have no uncertainty about the points we have already sampled. To see this, note that if $\mathbf{x}^* = \mathbf{x}_i$, then $\mathbf{r}$ is just the i^{th} column of

$\mathbf{R}$. Hence, $\mathbf{R}^{-1}\mathbf{r}$ is the i^{th} unit vector $\mathbf{e}_i$. It follows that

$$\mathbf{r}'\mathbf{R}^{-1}\mathbf{r} = \mathbf{r}'\mathbf{e}_i = \varphi\left(\mathbf{x}^* - \mathbf{x}_i\right) = \varphi\left(\mathbf{x}_i - \mathbf{x}_i\right) = 1 \tag{24}$$

$$\mathbf{1}'\mathbf{R}^{-1}\mathbf{r} = \mathbf{1}'\mathbf{e}_i = 1. \tag{25}$$

Substituting Eqs. (24) and (25) into Eq. (23), it follows that $s^2\left(\mathbf{x}_i\right) = 0$.

It will often be convenient for us to work with the square root of the mean squared error, $s = \sqrt{s^2\left(\mathbf{x}^*\right)}$. This provides a root mean squared error, or 'standard error,' for measuring uncertainty in our predictions.

A key difference between kriging and other basis function methods is that the other methods usually have no parameters in their basis functions. A possible exception is the parameter γ used in multiquadrics, but this parameter is rarely adjusted in any systematic manner. Moreover, the use of the Euclidean norm by many of the methods makes them sensitive to the units of measurement. For example, changing units from milimeters to meters could substantially change predictions. To avoid this problem, the standard procedure is to normalize all the data to the unit interval. While this certainly removes sensitivity to units of measurement, it also treats all variables as equally important—something that is almost never true. In constrast, kriging effectively uses a non-Euclidean distance metric given by

$$\text{distance}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \sum_{\ell=1}^{d} \theta_\ell \left|\mathbf{x}_{i\ell} - \mathbf{x}_{j\ell}\right|^{p_\ell} \tag{26}$$

Any change in the units can be absorbed by the θ_ℓ parameters. Moreover, the θ_ℓ and p_ℓ parameters can be adjusted to reflect the relative importance of each variable. The maximum likelihood estimation of these parameters is essentially a way of optimally 'tuning' the basis functions. This tuning is the main reason kriging often outperforms other basis-function methods in terms of prediction accuracy.

To see the importance of proper scaling, Figure 12 compares the performance of kriging and thin-plate splines on the Branin test function. Figure 12(a) shows the contours of Branin test function. Figure 12(b) shows the contours of a thin-plate spline fit using normalized data to 21 points (shown as spheres). Figure 12(c) shows the kriging predictor fit using normalized data with $p_1 = p_2 = 2$. The kriging predictor is clearly more accurate than the spline. In Figure 12(d) we do something special. In particular, we fit the thin-plate spline with the data scaled in a way suggested by the estimated kriging parameters. In particular, we scale the first variable to $\left[0, \sqrt{\widehat{\theta_1}}\right]$ and the second variable to $\left[0, \sqrt{\widehat{\theta_2}}\right]$. This has the effect of causing the thin-plate spline to use the same distance metric as kriging. As one can see, the resulting contours are somewhat better (i.e., more like those of the true function), demonstrating the importance of proper scaling.

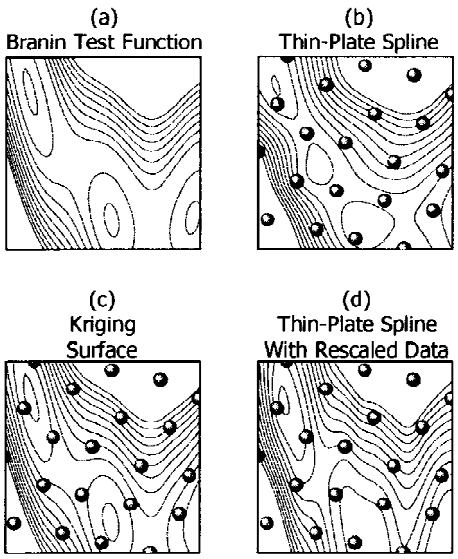

Figure 12. Comparison of the contours of the Branin test function; a thin-plate spline fit to the 21 points; a kriging surface; and a thin-plate spline fit using the same variable scaling as kriging.

5. Minimizing a statistical lower bound

The availability of a standard error in kriging immediately suggests the possibility of computing a 'statistical lower bound' on the function of the form $\widehat{y}(\mathbf{x}^*) - \kappa s(\mathbf{x}^*)$ for some value of κ. Why couldn't we then use this bound to develop a sort of nonrigorous branch-and-bound algorithm? As long as $\kappa > 0$, this would give us the desired property of putting some emphasis on searching in relatively unsampled regions where the standard error is higher. In the literature, this idea has been pursued by Cox and John (1997). In Figure 13 we explore whether or not this approach will allow us to find the global minimum of the deceptive Test Function #4.

In each iteration, we fit a kriging model, find the minimum of $\widehat{y}(\mathbf{x}^*) - 5s(\mathbf{x}^*)$, evaluate the function at that point, and then iterate. One would think that using the mean minus *five* standard errors would be a very conservative bound. But one must remember that $s(\mathbf{x}^*)$ is only an *estimate* of the possible error in the predictor. In our case, the initial points are quite deceptive, making the function appear very smooth and almost quadratic. As a result, the standard errors are very small. After sampling the next two points, however, it becomes clear that the function is more variable than it had seemed, and the standard errors become larger (see Iteration 3). Nevertheless, in the region between the first two points (which is where the global optimum lies) the lower bound is still above the current best function value. As a result, the search will not return to this region and the global optimum will not be found. Like a true branch-and-bound algorithm, the idea of minimizing a 'statistical

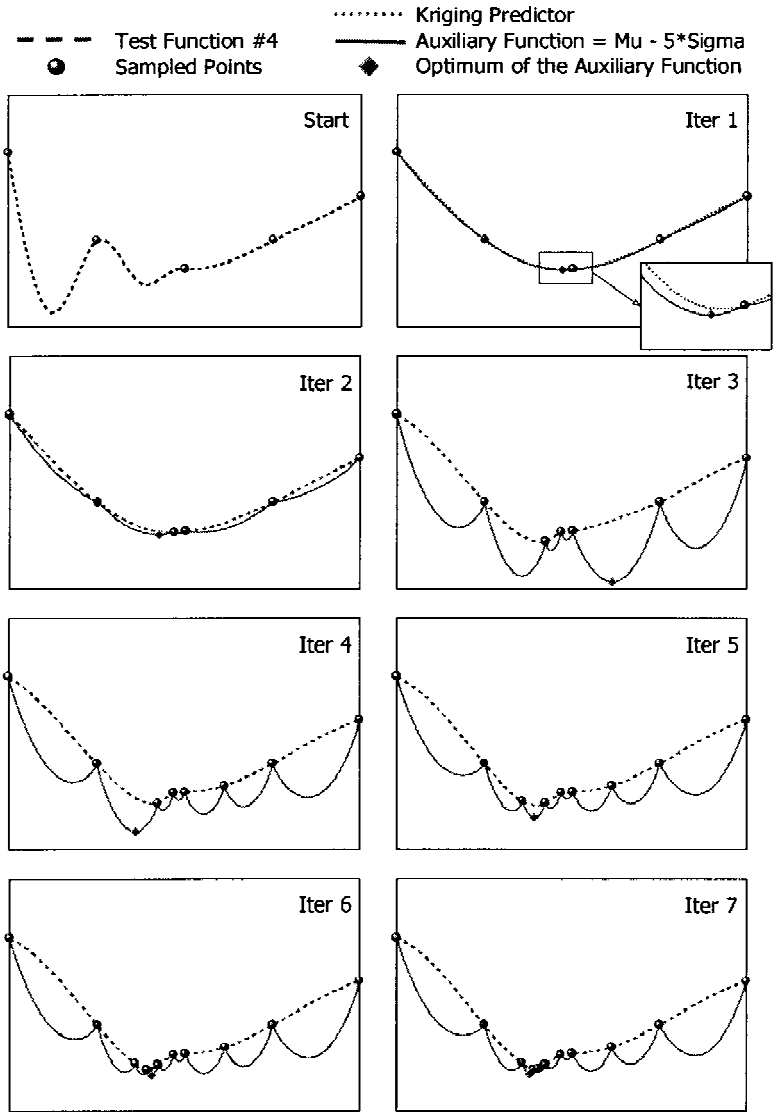

Figure 13. Selected iterations of Method 3 on a simple one-dimensional test function. In each iteration, we fit a kriging surface to the data, find the point that minimizes the predictor minus κ standard errors, and then evaluate the function at this point.

lower bound' will lead to the deletions of regions of the search space, and so the iterates will not be dense. But from the theorem of Torn and Zilinskas, the iterates *must* be dense if we are to guarantee convergence for general continuous functions.

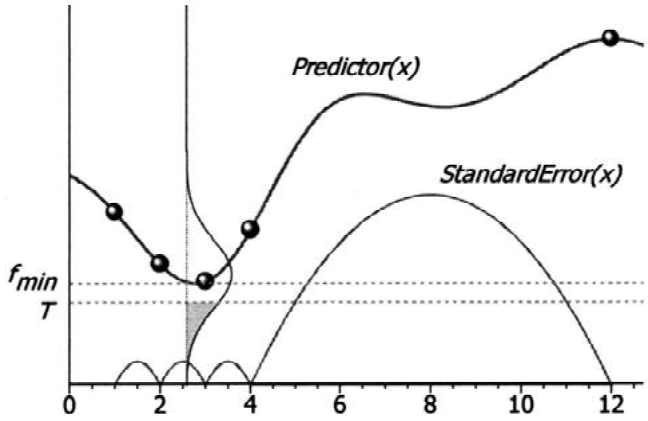

Figure 14. Using kriging, we can estimate the probability that sampling at a given point will 'improve' our solution, in the sense of yielding a value that is equal or better than some target T.

6. Maximizing the probability of improvement

In the literature, one of the most popular approaches for selecting an iterate is to find the point where the probability of improving the function beyond some target T is the highest. This concept is illustrated in Figure 14 (the standard error has been exaggerated for clarity). At any given point, we model our uncertainty about the function's value by considering this value to 'be like' the realization of a random variable $Y(\mathbf{x})$ with mean $\widehat{y}(\mathbf{x})$ and standard error $s(\mathbf{x})$. If we denote the current best function value as $f_{\min}$, then our target value for the improvement will be some number $T < f_{\min}$. The probability of improving this much (or more) is simply the probability that $Y(\mathbf{x}) \leqslant T$. Assuming the random variable is normally distributed, this probability is given by

$$\text{Prob Improvement} = \Phi\left(\frac{T - \widehat{y}(\mathbf{x})}{s(\mathbf{x})}\right) \tag{27}$$

where $\Phi(\cdot)$ is the Normal cumulative distribution function. This was the idea first put forward by Harold Kushner in 1964. Kushner's original algorithm was one dimensional, but Stuckman (1988), Perttunen (1991), Elder (1992), and Mockus (1994), have all heuristically extended the method to higher dimensions. Zilinskas (1992) introduced an axiomatic treatment of the method, which he calls the 'P-algorithm.'

The key advantage of using the probability of improvement is that, under certain mild assumptions, the iterates will be dense. Gutmann (2001) proves this result for Kushner's original one-dimensional algorithm (it is a special case of a more general convergence theorem). Intuitively, as the function is sampled more and more around the current best point, the standard error in this region becomes small.

As a result, the term

$$\frac{T - \hat{y}(\mathbf{x})}{s(\mathbf{x})}$$

becomes extremely negative (since we usually have $T < \hat{y}(\mathbf{x})$) and, hence, the probability of improvement given by Eq. (27) will be small. Eventually, the probability of improvement around the current best point becomes so small that we are driven to search elsewhere where the standard error is higher. This probability-of-improvement algorithm is 'Method 4' in the taxonomy of Figure 2.

Figure 15 explores how this method works for Test Function #4 when we set our target $T = f_{\min} - 0.25\,|f_{\min}|$, that is, when we search for an improvement of at least 25%. The method works just as expected. It starts around the suboptimal local minimum, but after sampling there a few times, it moves on to search more globally. By Iteration 11, the algorithm has found the basin of convergence of the global minimum.

The performance of Method 4 in Figure 15 is truly impressive. It would be quite natural if the reader, like so many others, became enthusiastic about this approach. But if there is a single lesson to be taken away from this paper, it is that nothing in this response-surface area is so simple. There always seems to be a counterexample. In this case, the difficulty is that Method 4 is extremely sensitive to the choice of the target T. If the desired improvement is too small, the search will be highly local and will only move on to search globally after searching nearly exhaustively around the current best point. On the other hand, if T is set too high, the search will be excessively global, and the algorithm will be slow to fine-tune any promising solutions. This sensitivity to the setting of the target is illustrated in Figure 16 which contrasts the status of the search after 11 iterations when using a target of 25% versus 1% improvement.

There are two ways to overcome this sensitivity. One way is to change the auxiliary function to something called 'expected improvement.' We will discuss this option in the next section. However, probably the best approach is to simply use several values of the target T corresponding to low, medium, and high desired improvement. This will lead to the selection of several search points in each iteration, causing us to search both locally and globally in each iteration. As a result, as soon as the global part of the search stumbles into the basin of convergence of the global minimum, the local part of the search will immediately begin to fine-tune the solution in the next iteration. Moreover, generating several search points per iteration allows one to take advantage of any parallel computing capabilities that may be available. We will refer to this approach as 'Enhanced Method 4'.

To illustrate this, let us return to the 'Start' panel of Figure 14 and see what happens if we find the point that maximizes the probability of improvement for *several* values of the target T. Now selecting any finite set of values for T must inevitably be somewhat arbitrary. In my experience, however, the following procedure seems to work well. First, find the minimum of the surface using multistart

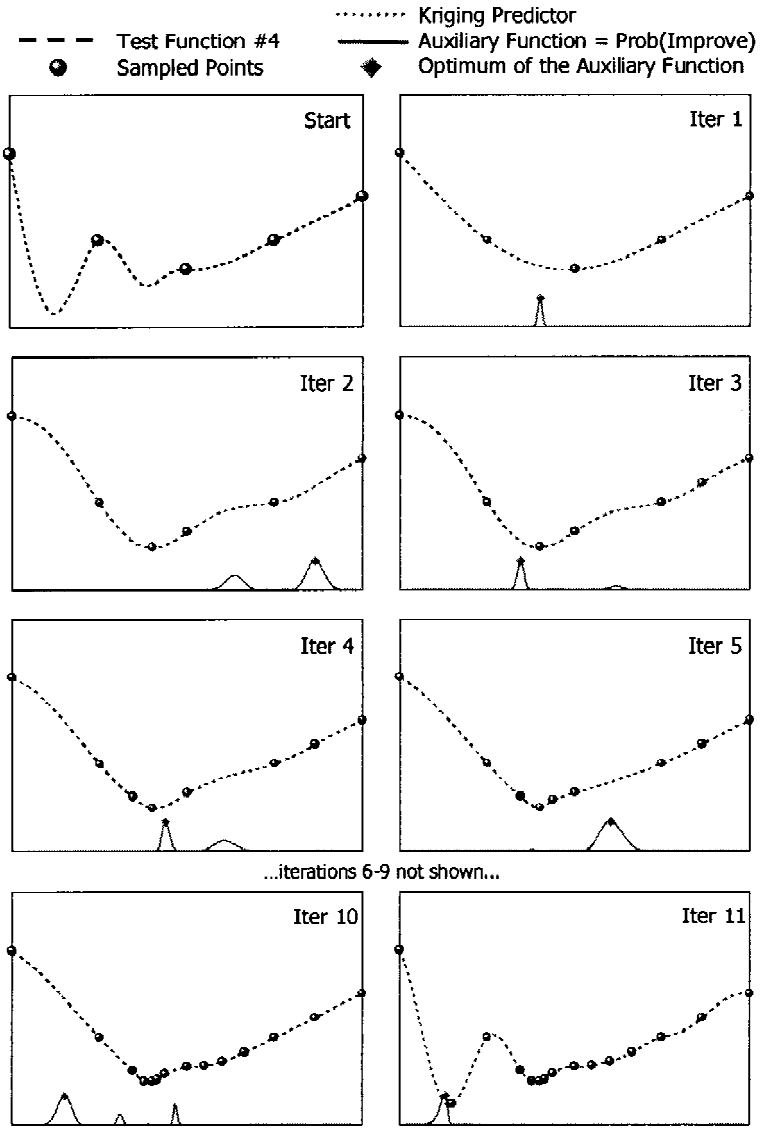

Figure 15. Illustration of Method 4 on a simple one-dimensional test function. In each iteration, we fit a kriging surface to the data, find the point that maximizes probability of improvement (defined as exceeding some target T), and then evaluate the function at this point. In this example, the target T was set 25% below the minimum of the surface.

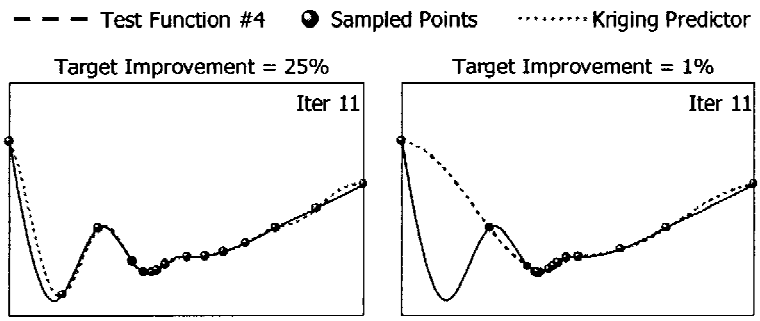

Figure 16. Method 4 is sensitive to the desired improvement. On the left the target for improvement was set 25% below the minimum of the surface, and the search is quite global. On the right, the target is set to 1% below the minimum of the surface, and the search is excessively local in nature.

Table 1. The 27 values of α used to compute function-value targets in Enhanced Method 4.

Target Number	α	Target Number	α	Target Number	α
1	0.0	10	0.07	19	0.25
2	0.0001	11	0.08	20	0.30
3	0.001	12	0.09	21	0.40
4	0.01	13	0.10	22	0.50
5	0.02	14	0.11	23	0.75
6	0.03	15	0.12	24	1.00
7	0.04	16	0.13	25	1.50
8	0.05	17	0.15	26	2.00
9	0.06	18	0.20	27	3.00

and call this value $s_{\min}$. Also find the minimum and maximum objective function value at the sampled points; call these $f_{\min}$ and $f_{\max}$. Now construct targets using $T = s_{\min} - \alpha(f_{\max} - f_{\min})$ using the 27 values of α shown in Table 1. (When $\alpha = 0$, the point that maximizes the probability of improvement is the same as the point that minimizes the surface.)

Figure 17 shows what happens when we use these 27 target values. In the figure, each diamond shows the result of maximizing the probability of improvement for a given target value T. The horizontal coordinate of the diamond is the x coordinate of the point that maximizes the probability of improvement. The vertical coordinate corresponds to the 'target number' from Table 1 and is shown on the right-hand scale. As one can see, targets near the current minimum (e.g., target number 1) result in search points close to the current best point. As the target level T decreases (target *number* increases), the point maximizing the probability of improvement moves away from the current best point towards a more unexplored part of the

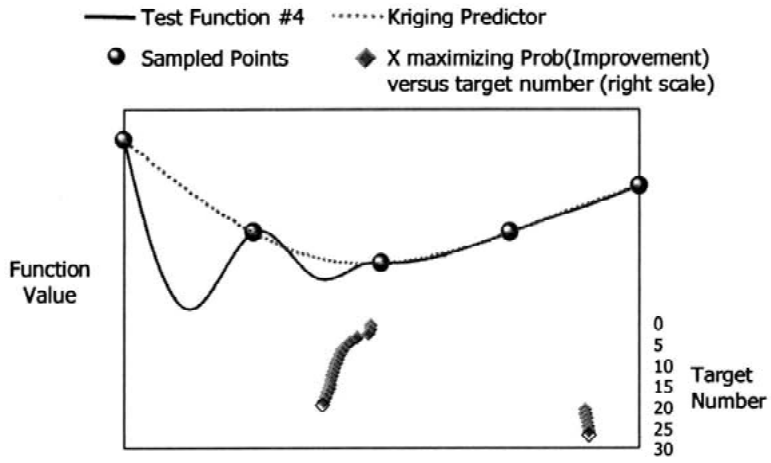

Figure 17. Location of the point that maximizes the probability of improvement (diamonds) as a function of the amount of improvement desired, given by the 'target number' (right hand scale).

space. The diamonds corresponding to targets 1–20 cluster in one part of the space, whereas those for targets 21–27 cluster in a different areas. In practice, it makes sense to sample only one point from each region. What we do, therefore, is to cluster the points and, from each cluster, take the point associated with the *lowest* target. In Figure 17, these points are shown as open, unfilled diamonds. Choosing the point associated with the lowest target makes the search more global and insures that the sampled points will be dense.

In Figure 18 we show what happens if we iterate this process on Test Function #4. In this figure, we use vertical, dotted lines to show the x values corresponding the new search points chosen via the clustering procedure. Thus, the two lines shown for Iteration 1 correspond to the two open diamonds in Figure 17. Because the method searches both globally and locally, we find the basin of convergence of the global optimum much sooner (in Iteration 2). Moreover, because we sample between one and three points per iteration, some speed up from parallel computation is possible.

Enhanced Method 4, like all of the methods described so far, is not limited to one-dimensional problems. For example, Figure 19 shows how the method performs on the two-dimensional Branin test function. By the sixth iteration the method has sampled near all three global minima (they are tied for the best value). The number of points sampled per iteration varies from 1 to 5.

Some technical details of Enhanced Method 4 are given the Appendix. These details include the method for generating starting points for maximizing the probability of improvement via multistart, and also the method for clustering the resulting solutions.

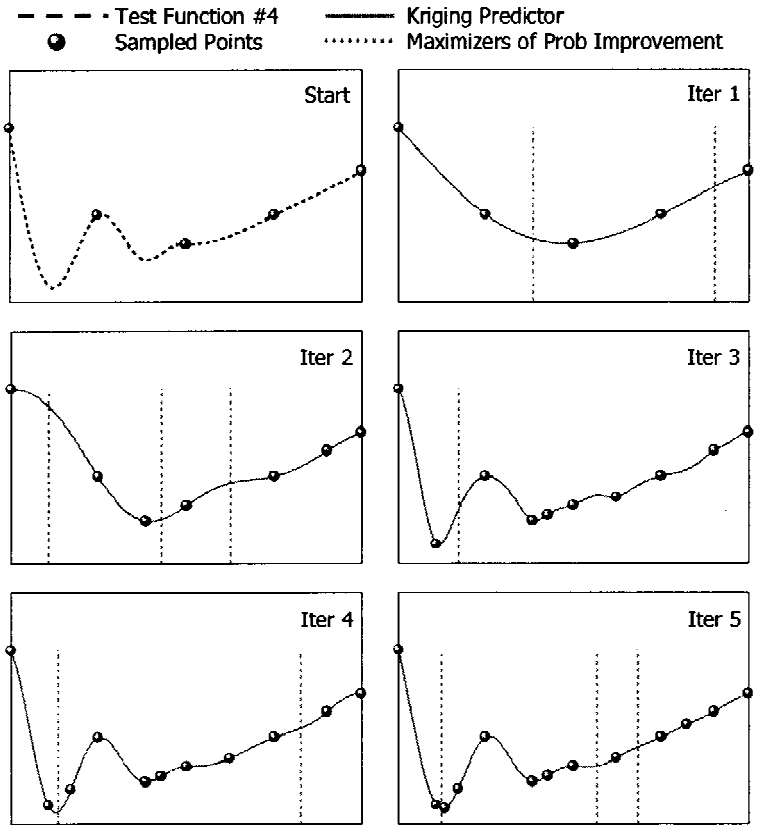

Figure 18. The first five iterations of enhanced method #4 applied to test function #4.

As an aside, note that we could also have constructed an 'Enhanced Method 3' by employing several values of the constant κ used to compute the 'statistical lower bound' $\widehat{y}(\mathbf{x}^*) - \kappa s(\mathbf{x}^*)$. This would be very similar in spirit to using several values of the target T in Method 4. There is no need to do this, however, because it produces exactly the same search! To see this, note that maximizing the probability of improvement is the same as maximizing

$$\Phi\left(\frac{T - \widehat{y}(\mathbf{x})}{s(\mathbf{x})}\right) \tag{28}$$

which is the same as minimizing

$$\frac{\widehat{y}(\mathbf{x}) - T}{s(\mathbf{x})}. \tag{29}$$

Now suppose that the minimum of the above ratio occurs at point $\mathbf{x}^*$ and that, at this point, the ratio is κ. If we then set

$$T = \widehat{y}\left(\mathbf{x}^*\right) - \kappa s(\mathbf{x}^*), \tag{30}$$

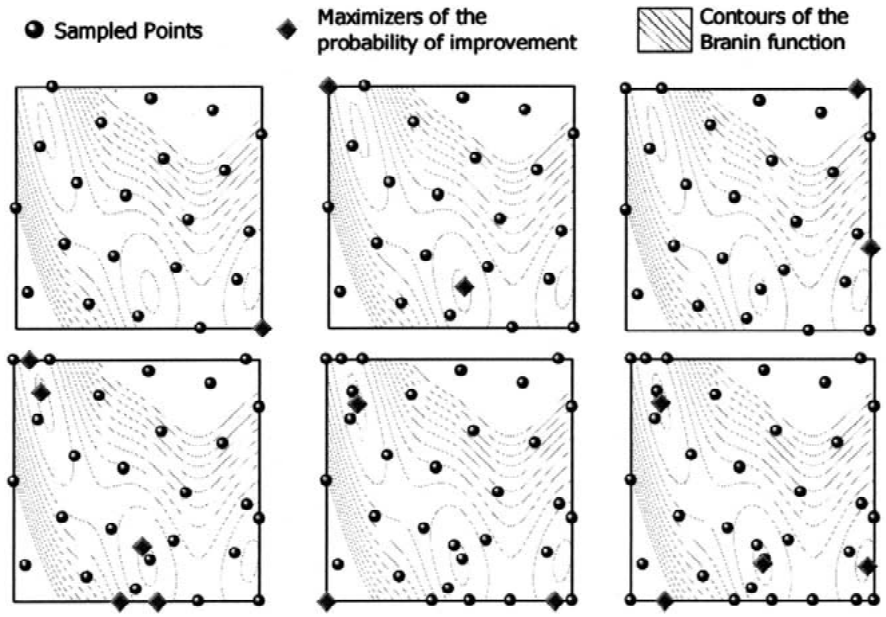

Figure 19. First six iterations of enhanced method #4 on the Branin function.

then the point $\mathbf{x}^*$ must also minimize $\widehat{y}(\mathbf{x}^*) - \kappa s(\mathbf{x}^*)$. To see this, suppose there were some point $\mathbf{x}'$ such that

$$\widehat{y}\left(\mathbf{x}'\right) - \kappa s(\mathbf{x}') < T. \tag{31}$$

It would then follow that

$$\frac{\widehat{y}\left(\mathbf{x}'\right) - T}{s\left(\mathbf{x}'\right)} < \kappa = \frac{\widehat{y}\left(\mathbf{x}^*\right) - T}{s\left(\mathbf{x}^*\right)}, \tag{32}$$

which would contradict our assumption that point $\mathbf{x}^*$ minimized the ratio on the right hand side. Thus we see that there is an isomorphism between choices of T in Method 4 and choices of κ in Method 3. Matching pairs of T and κ give the same optimum of the auxiliary problem. Hence, using *all* possible values of T in Method 4 results in the same search as using *all* possible values of κ in Method 3.

Note, however, that using *one* value of T is *not* the same as using *one* value of κ. The reason is that the value of κ that corresponds to a given T depends not only on T but also on the minimium value of this ratio in Eq. (29), which will change from iteration to iteration.

Using several targets is probably the best way to resolve the sensitivity of Method 4 to the choice of the target improvement (or the sensitivity of Method 3 to the number of standard deviations κ). As mentioned earlier, however, there is another way that has also attracted a great deal of attention, based on the concept of 'Expected Improvement.' We consider this next.

7. Maximizing Expected Improvement

The 'expected improvement approach,' as the name suggests, involves computing how much improvement we expect to achieve if we sample at a given point. As before, let $Y(\mathbf{x})$ be a random variable describing our uncertainty about the function's value at a point $\mathbf{x}$, and assume that $Y(\mathbf{x})$ is normally distributed with mean and variance given by the kriging predictor, that is, by $\widehat{y}(\mathbf{x})$ and $s^2(\mathbf{x})$. If the current best function value is $f_{\min}$, then we will achieve an improvement of I if $Y(\mathbf{x}) = f_{\min} - I$. The likelihood of achieving this improvement is given by the normal density function

$$\frac{1}{\sqrt{2\pi}s(\mathbf{x})} \exp\left[-\frac{(f_{\min} - I - \widehat{y}(\mathbf{x}))^2}{2s^2(\mathbf{x})}\right] \tag{33}$$

The expected improvement is simply the expected value of the improvement found by integrating over this density:

$$E(I) = \int_{I=0}^{I=\infty} I \left\{\frac{1}{\sqrt{2\pi}s(\mathbf{x})} \exp\left[-\frac{(f_{\min} - I - \widehat{y}(\mathbf{x}))^2}{2s^2(\mathbf{x})}\right]\right\} \mathrm{d}I \tag{34}$$

Using integration by parts, one can show that

$$E(I) = s(\mathbf{x})\left[u\Phi(u) + \phi(u)\right] \tag{35}$$

where

$$u = \frac{f_{\min} - \widehat{y}(\mathbf{x})}{s(\mathbf{x})}$$

and where Φ and ϕ are the normal cumulative distribution function and density function, respectively. In Method 5, we will fit a kriging model, find the point that maximizes expected improvement, evaluate the function at this point, and iterate.

The appeal of the expected improvement approach is three-fold. First, it avoids the need specify a desired improvement (i.e., the target T of the previous section). Second, Locateli (1997) has proved that under certain assumptions the iterates from this method are dense. Third, it provides a simple stopping rule: 'stop when the expected improvement from further search is less than some small positive number'. In the literature, this method has been pursued by Schonlau et al. (1997), and by Sasena (2000). The performance of this method on Test Function #4 is explored in Figure 20.

As guaranteed by Locateli's proof, the expected improvement method does find the global minimum. But in this case it takes a long time to do so. The reason is that the initial sample is highly deceptive, leading to very small estimates of the standard error. As a result, only points that are close to the current best point have high expected improvement. It requires fairly exhaustive search around the initial best point before the algorithm begins to search more globally.

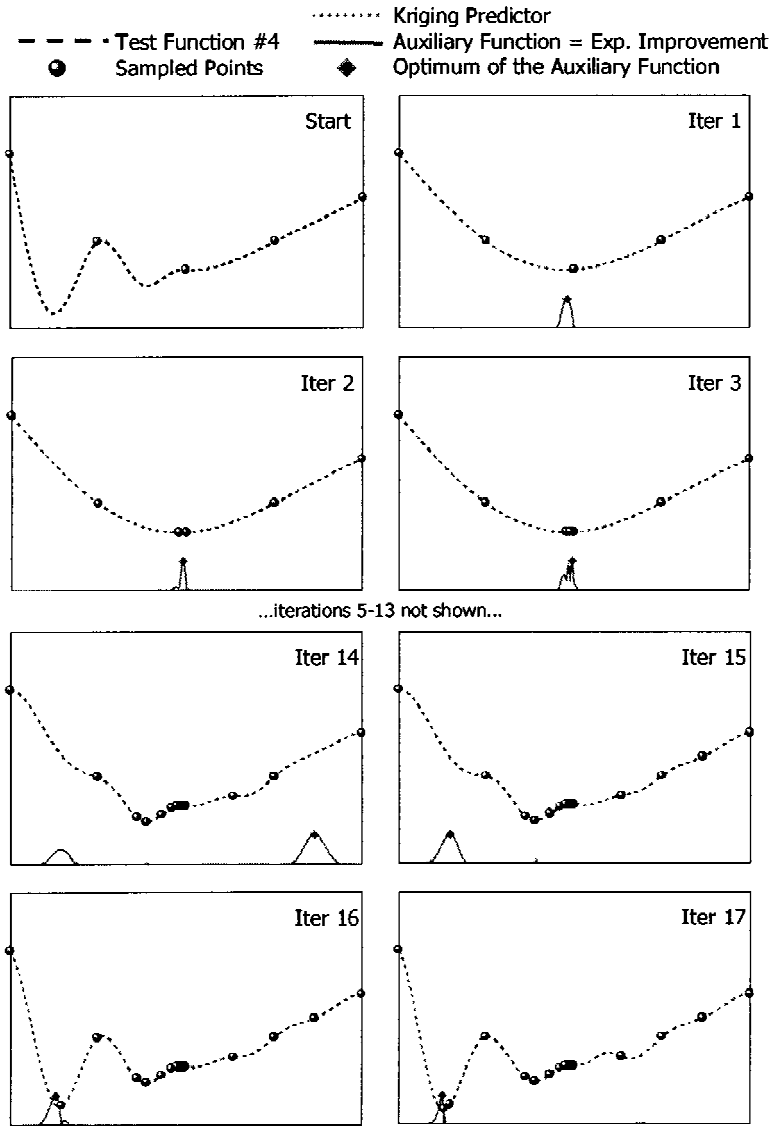

Figure 20. Illustration of Method 5 on a simple one-dimensional test function. In each iteration, we fit a kriging surface to the data, find the point that maximizes expected improvement, and evaluate the function at this point.

This potential for deception is a fundamental weakness of Methods 3–5. All of these methods rely on the standard error of the kriging predictor to force the algorithm to go back and explore regions where the sampled points are sparse. This is certainly better than completely ignoring the potential error in the surface. But all of these methods treat the estimated standard error as if it is correct, which will not always be the case. Test Function #4 is a good example where the standard

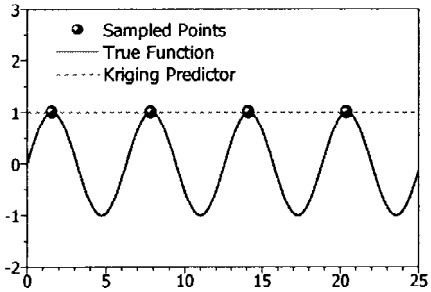

Figure 21. The sample used to construct the initial surface can be extremely deceptive, leading to gross under-estimation of the error in the kriging predictor. This can lead to poor peformance in 'two stage' approaches (Methods 1–5 in the taxonomy) that first fit a response surface and then use the surface to select the next iterate.

error may be greatly underestimated. An even more extreme case of deception is shown in Figure 21. In this case, the true function is the sine function, and we have unluckily sampled it at all the crests. The kriging predictor fitted to this data would have $\mu = 1$ and $\sigma^2 = 0$ and would predict that the probability of improvement and the expected improvement are equal to zero everywhere.

In short, the fundamental flaw of Methods 3–5 is that they are *two-stage* methods. In stage 1, the kriging surface is fit to the observed data, and all the relevant parameters are estimated. In the second stage, these parameter estimates are taken as correct and a calculation is done to determine where to search. Two-stage methods can be deceived when the initial sample is sparse and gives a highly misleading view of the function. Because the misleading view is taken as 'correct' in stage 2, the algorithm may stop prematurely or become excessively local in its selection of search points. To avoid this pitfall, we must avoid estimating the parameters of the kriging model based *only on the observed sample*. While this may seem impossible, it is precisely what we do in the 'one-stage' methods we consider next.

8. One-stage approach for goal seeking

In this section, we assume that we have a target value or 'goal' for the objective function. That is, instead of minimizing the function, we merely want to achieve a value f^* which is considered to be desirable. For example, a goal might be set by benchmarking to competitive products. In this section we will explore a 'one-stage approach' for such goal-seeking problems. In the next section, we extend this method to the case of true optimization.

In a one-stage approach, we posit a hypothesis about the location of the point that achieves the goal f^* and use the machinery of response surfaces to measure the 'credibility' of this hypothesis. More specifically, suppose we hypothesize that the goal f^* is achieved at a point $\mathbf{x}^*$. To evaluate this hypothesis, we compute the likelihood of the observed data *conditional* upon the assumption that the surface

goes through the point $(\mathbf{x}^*, f^*)$. This conditional likelihood is:

$$\frac{1}{(2\pi)^{\frac{n}{2}}\left(\sigma^2\right)^{\frac{n}{2}}|\mathbf{C}|^{\frac{1}{2}}}\exp\left[\frac{-(\mathbf{y}-\mathbf{m})'\,\mathbf{C}^{-1}(\mathbf{y}-\mathbf{m})}{2\sigma^2}\right] \tag{36}$$

where $\mathbf{m}$ and $\mathbf{C}$ are the conditional mean and correlation matrix:

$$\mathbf{m} = \mathbf{1}\mu + \mathbf{r}(f^* - \mu) \tag{37}$$

$$\mathbf{C} = \mathbf{R} - \mathbf{r}\mathbf{r}' \tag{38}$$

Note that the value of $\mathbf{x}^*$ comes into play through the vector $\mathbf{r}$ of correlations between $\mathbf{x}^*$ and the n sampled points.

When using the conditional log-likelihood to evaluate the hypothesis that the surface passes through $(\mathbf{x}^*, f^*)$, we also optimize the kriging parameters μ, σ^2, θ_ℓ and p_ℓ $(\ell = 1, .., d)$ to maximize the conditional likelihood. That is, the parameters are adjusted to make the hypothesis seem as likely as possible. The resulting optimized likelihood can be considered to be a measure of the 'credibility' of our hypothesis.

The next iterate is the value of $\mathbf{x}^*$ that maximizes the measure of credibility. Mechanically, it is found by globally maximizing the conditional log-likelihood over both $\mathbf{x}^*$ *and* the kriging parameters μ, σ^2, θ_ℓ and p_ℓ $(\ell = 1, .., d)$. As before, one can concentrate out the parameters μ and σ^2 and thereby express the conditional log-likelihood as a function of only $\mathbf{x}^*$ and the correlation parameters θ_ℓ and p_ℓ $(\ell = 1, .., d)$. The key thing to note is that the next iterate is *not* based on parameters obtained by fitting a surface to the observed data alone—parameters that can be greatly in error if the initial sample is sparse and misleading.

To illustrate this concept, Figure 22 explores two hypothesized locations where Test Function #4 might reach a goal f^* represented as a horizontal line. For both cases, we show the optimized conditional log-likelihood as well as the best-fitting kriging surface through the hypothesized point and the data. I think most readers would agree that the graph on the right, which has the higher credibility value (conditional log-likelihood), is indeed the more believable hypothesis.

The performance of Method 6 on our tricky Test Function #4 is shown in Figure 23. In this example we have 'cheated' and set the goal f^* equal to the global minimum—something that we could obviously not do in practice. The search finds the basis of the global minimum after only seven iterations. The perforance of Method 6 on Test Function #4 is comparable to that of Enhanced Method #4, but unlike that method, Method 6 can handle extremely deceptive problems like that shown in Figure 21.

Of course, in Method 6 we have assumed we know the optimal function value f^* or an appropriate goal to seek. But what should we do if we did not know the minimum or have a clear goal? We take up this topic in the next section.

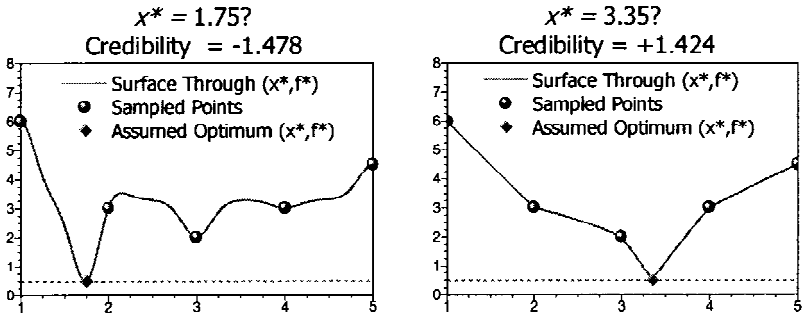

Figure 22. In Method 6, the credibility of the hypothesis that the surface passes through (x^*, f^*) is based on the likelihood of the data conditional upon the surface passing through this point. The figure compares hypotheses for two possible values of x^*, both with the same value of f^*.

9. One-stage approach for optimization

When a search goal f^* is not available, we can stil use the basic idea of Method 6 for optimization. We simply compute *several* search points using *several* values of $f^* < f_{\min}$. On the face of it, this would seem impractical. After all, if one tries many values of f^*, it would seem that one would get many search points, which might be wasteful. As it turns out, however, the search points computed for the different values of f^* will tend to cluster, just as they did earlier in the enhanced version of Method 4.

Gutmann (2001) has implemented this approach using thin-plate splines instead of kriging. This simplifies things considerably since the thin-plate splines have no unknown parameters—like the θ_ℓ and p_ℓ parameters in kriging—that need to be estimated. Furthermore, instead of trying several f^* values in a single iteration and clustering the resulting points, Gutmann cycles through five values for f^* ranging from low to high over the course of five successive iterations. These simplifications yield a quite tractable method, and he reports excellent numerical results on the Dixon–Szego test functions. Moreover, Gutmann has been able to prove the convergence of this approach subject to certain mild restrictions on the smoothness of the function and the values of f^* that are used. It is not clear whether or not this convergence proof applies to Method 6, in which the value of f^* is fixed and known.

We have implemented Method 7 using kriging in much the same way as we implemented the enhanced version of Method 4. That is, we use the 27 target values for f^* given by Table 1. For each of these targets, we search for the global maximum of the conditional loglikelihood, searching over both $\mathbf{x}^*$ *and* the correlation parameters θ_ℓ and p_ℓ. The global optimization is done using multistart, and the best solutions for the 27 target values of f^* are clustered, as detailed in the Appendix. Figure 24 shows the result of applying Method 7 to Test Function #4. Method 7 takes 15 evaluations to find the basin of the global minimum, compared to only 12

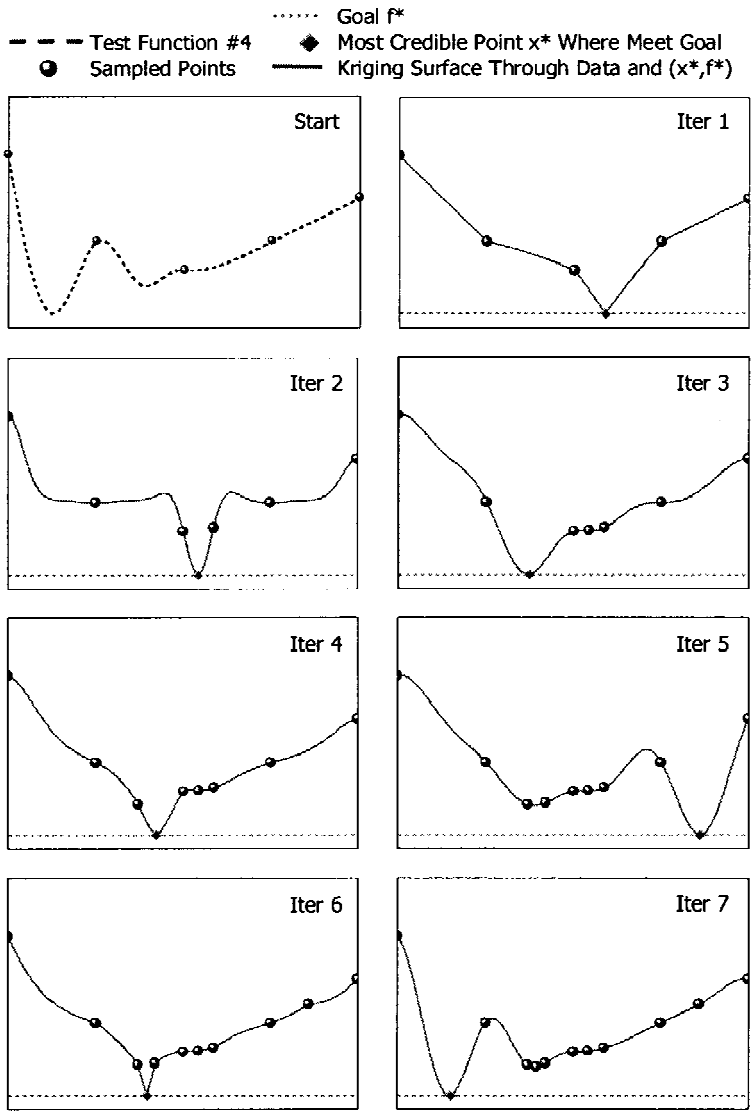

Figure 23. Illustration of Method 6 on a simple one-dimensional test function. We assume we want to find a point where the objective function achieves a 'goal' value of f^*. In each iteration, we find the point x^* with the property that the surface going through the data and the point $(x^*,\ f^*)$ is as 'credible' as possible. Here 'credibility' is measured by the log-likelihood of the data conditional upon the surface passing through $(x^*,\ f^*)$.

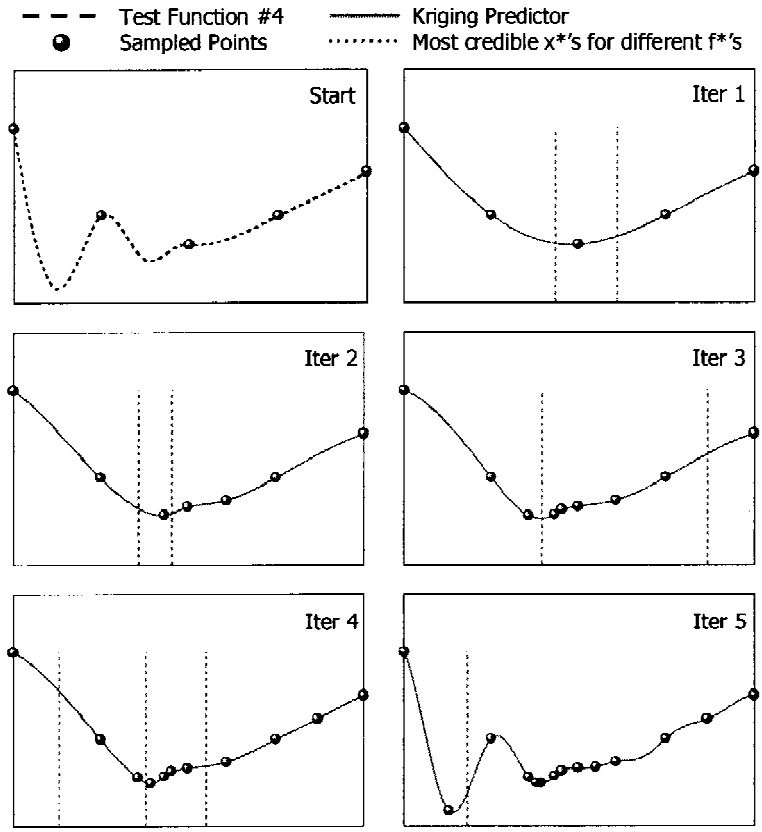

Figure 24. Performance of Method 7 on Test Function #4.

for Method 6 and only 10 for the Enhanced Method 4. This is the price we pay for the additional robustness.

10. Conclusions

Let us now summarize the findings from our tour of the seven methods. Methods 1 and 2 are the simplest approaches. One simply fits a surface, finds the minimum of the surface, evaluates the function at the surface minimum, and then iterates. We showed that, when this simple approach is implemented using quadratic response surfaces (Method 1), it can easily fail. Much more interesting is the case when the approach is implemented using interpolating surfaces such as splines or kriging (Method 2). Simple numerical examples shown that this method can easily miss the global minimum, but we spent a good deal of space discussing the merits of the approach for local optimization. This discussion showed that to insure convergence to a critical point, one may need to force the gradient of the surface to match the gradient of the function whenever the search stalls. Whether

or not this is sufficient to prove convergence to a critical point was identified as an open research question. By employing both gradient matching *and* a trust region approach, Alexandrov et. al. (2000) have developed a locally convergent method. Furthermore, Booker et.al. (1999) have shown how response surfaces can be used to accelerate a derivative-free method of local optimization.

We then moved on to discuss methods that attempt to make the search global by exploiting kriging's ability to estimate potential error in its predictions. The first of these methods (Method 3) determines the next iterate by minimizing a 'statistical lower bounding function', based on the predictor minus several standard errors. Unfortunately, this method was found to have the fatal flaw that the iterates would not be dense, and hence the search could fail to find the global minimum. We then considered Method 4 in which the next iterate maximizes the (estimated) probability that the function value at a point will be better than a target T. This method was found to be convergent, but sensitive to the choice of T. However, one can finesse this problem by computing several search points per iteration, using several values of T. The resulting 'Enhanced Method 4' appears to be a highly promising approach. Finally, we considered Method 5 in which the next iterate maximizes the expected improvement from sampling at a point. Methods 3–5 all rely upon the standard error computed in kriging and can perform poorly if initial sample is highly deceptive (this is especially true of Method 5). Deceptive samples can cause the kriging standard error to underestimate the true error in the predictor and, as a result, Methods 3–5 may converge prematurely or slowly. The use of several targets in 'Enhanced Method 4' seems to reduce the negative impact of a deceptive initial sample—which is one reason why we consider this approach so promising.

We then turned our attention to 'one-stage' methods that can avoid being deceived by a deceptive initial sample. These methods use the mathematical machinery of response surfaces to directly evaluate hypotheses about the location of the optimum. In Method 6 we assumed that we merely want to achieve a known goal f^* for the objective function. The next iterate was the point x^* where, in a sense that we made precise, it is 'most credible' that the function obtains the value f^*. This method was found to converge quickly to the goal. In Method 7, we assumed that we want to find the global minimum of the objective function and that the minimal function value is *not* known in advance. This case is handled by computing several search points using several values of f^*, just as we used several values of T in Enhanced Method 4. Gutmann (2001) reports excellent numerical results for a spline-based implementation of Method 7 and proves the convergence of the method.

Overall, three methods stand out as most promising: Enhanced Method 4, Method 6, and Method 7. Extending these methods to constrained problems is an open and important challenge for research. Some progress has already been made in handling constraints for the expected improvement approach (see Schonlau et al. (1997) and Booker et al. (1999)). Enhanced Method 4 is the most tractable

method and extending it to handle nonlinear constraints is an important direction for research. Methods 6 and 7 are computationally very intensive if kriging is used for the surface, so adding constraints will be more difficult. However, constrained versions of Methods 6 and 7 may be possible using non-kriging surfaces such as splines.

In any case, the field is clearly ripe for more work, both in developing efficient implementations of the most promising methods, empirical work to compare the performance of these approaches to competing methods, and theoretical work on convergence properties.

Acknowledgements

Most of the ideas in this paper have taken shape as part of a long collaboration with William Welch of the University of Waterloo and Matthias Schonlau of the Rand Corporation. Michael Powell and Hans Gutmann helped by sharing their work on Method 7 (using thin-plate splines) long before it was published. Michael Powell also provided code for computing thin-plate splines which we used to produce Figure 12.

Appendix

Methods 1–7 all find the next search points by globally maximizing or minimizing an auxiliary function (e.g., the probability of improvement). We solve these auxiliary problems using multistart. In the Enhanced Method 4 and in Method 7, the auxiliary function is optimized for several values of a target T and the resulting solutions are clustered in order to determine the next search points. This Appendix gives the details of the multistart and clustering procedures.

Our method for computing starting points is based on finding the midpoints of line segments connecting pairs of sampled points. Now there are $n(n-1)/2$ such pairs, and hence the same number of midpoints. But many of these midpoints are closer to other sampled points, or to other midpoints, than they are to the two endpoints that generate them. The procedure we describe next essentially 'weeds out' some of these 'redundant' midpoints leaving a smaller set of points. An example of such a set of starting points is shown in Figure 25.

Here is the procedure. First, compute the distance between all pairs of sampled points and sort the pairs in ascending order by this distance. Now consider each of the pairs in the sorted order (i.e., starting with the closest pair and continuing on to the most distant pair). For each pair, compute the midpoint of the line segment between the pair. Next, find the closest point to this midpoint, considering all sampled points *and* any starting points that have already been generated. Skip this midpoint if there is a point that is closer to the midpoint than its 'parents' (the pair of points from which the midpoint was computed); otherwise, include this

Table 2. Numerical illustration of the clustering procedure described in Appendix 2.

Target (i)	x_1	x_2	Group	Δ_i	Criterion
1	0.12387	0.81828	1	0.55624	N.A.
2	0.54273	0.15242	2	0.29691	100.00000
3	0.96242	0.16529	3	0.00403	73.62717
4	0.96712	0.16206	3	0.00345	1.16868
5	0.96910	0.15760	3	0.00287	1.20250
6	0.97003	0.15365	3	0.00215	1.33750
7	0.97060	0.15067	3	0.00174	1.23033
8	0.97102	0.14824	3	0.00136	1.28500
9	0.97129	0.14634	3	0.00132	1.02800
10	0.97154	0.14449	3	0.00106	1.24573
11	0.97170	0.14300	3	0.57886	0.00183
12	0.16566	0.00000	4	0.00052	1121.40628
13	0.16493	0.00000	4	0.00043	0.00089
14	0.16432	0.00000	4	0.00040	0.83562
15	0.16375	0.00000	4	0.00040	0.80610
16	0.16318	0.00000	4	0.00077	0.52294
17	0.16209	0.00000	4	0.00165	0.46581
18	0.15975	0.00000	4	0.00141	1.17588
19	0.15776	0.00000	4	0.00121	1.16374
20	0.15605	0.00000	4	0.00197	0.61511
21	0.15327	0.00000	4	0.00154	1.27523
22	0.15109	0.00000	4	0.00274	0.56331
23	0.14722	0.00000	4	0.00172	1.59259
24	0.14479	0.00000	4	0.00228	0.75232
25	0.14156	0.00000	4	0.00132	1.72727
26	0.13969	0.00000	4	0.00154	0.85780
27	0.13751	0.00000	4	N.A.	1.16578

midpoint in the list of starting points. Continue in this way until all pairs of points have been processed.

Now suppose we use multistart to find the global optimum of the auxiliary function for different values of the target T in either Enhanced Method 4 or Method 7. We use 27 different values of T, calculated using the desired improvement values in Table 2. For any desired improvement α, the corresponding target value is $T = s_{\min} - \alpha(f_{\max} - f_{\min})$, where f_{min} and f_{max} are the minimum and maximum sampled function values and where s_{min} is the minimum value of the response surface. We will now describe how we cluster the 27 points resulting from solving these auxiliary problems.

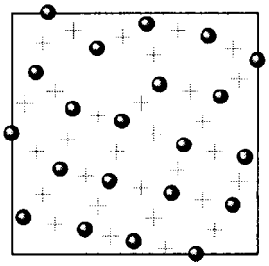

Figure 25. An example of starting points computed using the procedure of the Appendix. The spheres are the sampled points, and the crosses are the starting points for multistart.

Before continuing, the reader should know that clustering is more of an art than a science and the procedure we are about to describe is somewhat *ad hoc*. The procedure has been tuned to work well on a small set of test problems, but it will hopefully give reasonable performance on most problems. For concreteness, we will illustrate the procedure using iteration 6 of Enhanced Method #4 on the Branin function—that is, the lower right plot in Figure 19.

The results of optimizing the auxiliary function for the 27 values of T are shown in Table 2. In this table, the first column is the target number and the next two columns give the (x_1, x_2) coordinates of the point that maximized the auxiliary function for this target number. Note that the (x_1, x_2) data have been scaled so that each variable lies in the unit interval. The fourth column gives the group number (we will describe its calculation later). The fifth column is the root mean squared distance to the next point in the table, which we will denote by Δ_i for compactness. If we let d be the number of dimensions, and if we let $(x_{i1}, \ldots, x_{id})$ be the i^{th} point in the table, then this distance is:

$$\Delta_i = \sqrt{\frac{\sum_{k=1}^{k=d}(x_{ik} - x_{i+1,k})^2}{d}} \tag{39}$$

We initialize the clustering by assigning point 1 to group 1. We then sequentially consider points 2 through 27 and decide whether they should be classified in the same group as the previous point, or whether they should start a new group. We make this decision with the help of a criterion shown in the sixth column. In particular, we put point i in the same group as point $i-1$ if the criterion is less than 12; otherwise we start a new group. Here is how the criterion is computed:

- If Δ_i>0.1 and Δ_{i-1}>0.1, then set the criterion to 100 so that we start a new group. In this case, point i is quite distant from both point $i-1$ and point $i+1$ and should be in its own cluster.
- Otherwise, if Δ_i>0.0005, then set the criterion to Δ_{i-1}/Δ_i. In this case, the criterion will be high if the distance from point $i-1$ to point i is high but the distance between point i and point $i+1$ is low—a condition that indicates

that point i should be grouped with the points that follow it and not the ones that preceed it.

- Otherwise, if $i \geqslant 3$ and Δ_{i-1}>0.0005, then set the criterion to $\Delta_{i-1}/\max(\Delta_{i-2}, 0.0005)$. In this case, the criterion will be high if the distance from $i-1$ to i is much greater than the distance from $i-2$ to $i-1$. This, in turn would suggest that points $i-2$ and $i-1$ are in the same group but point i is so different from point $i-1$ that it should start a new group.
- Otherwise, if $i = 2$ and Δ_1>0.1 and Δ_2<0.0005, then set the criterion to 100 to signal the need for a new group. In this case point 2 is very different from point 1.
- If none of the above conditions are satisfied, set the criterion to zero so that we do not start a new group.

At this point we will have assigned a group to the point generated using each of the 27 targets. For each group, we take the point associated with the highest target number as a 'group representative' to be sampled on the next iteration.

This clustering procedure usually works very well. However, in implementing Method 7, I have found one strange failure mode for the clustering procedure, and it is necessary to correct this problem when it occurs. In particular, sometimes a later group (e.g., group 3) will be essentially the same as an earlier one (e.g., group 1). To handle this, after computing the list of group representatives as above, I scan them from group 2 to group n_{group}, computing the root-mean-squared distance of each point to all the previous group representatives. As before, in computing this distance, all the variables are normalized to lie on the unit interval. If a point is within 0.03 of a previous group representative in terms of root-mean-squared distance, then I skip it.

References

Alexandrov, N.M., Lewis, R.M., Gumbert, C. R., Green, L.L., and Newman, P.A. Optimization with variable-fidelity models applied to wing design. In *Proceedings of the 38th Aerospace Sciences Meeting & Exhibit*, January 2000. AIAA Paper 2000-0841.

American Institute of Aeronautics and Astronautics. *Proceedings of the 8th AIAA / USAF / NASA / ISMMO Symposium of Multidisciplinary Analysis & Optimization*, September 2000. Available on CD-ROM from the AIAA at http://www.aiaa.org.

Booker, A. J. Examples of surrogate modeling of computer simulations. Presented at the ISSMO/NASA First Internet Conference on Approximations and Fast Reanalysis in Engineering Optimization, June 14-27, 1998. Conference abstract available at http://www.blarg.net/~mandoid/webcon/. Full text in pdf format available by sending email to andrew.j.booker@boeing.com.

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

Cox, D. D and John S. (1997). SDO: A statistical method for global optimization. In N. Alexandrov, and M.Y. Hussaini, editors, *Multidisciplinary Design Optimization: State of the Art*, 315–329. SIAM, Philadelphia.

Dennis, J.E. Jr. and Torczon, V. (1991). Direct search methods on parallel machines. *SIAM Journal of Optimization*, 1, 448–474.

L.C.W. Dixon and G.P. Szego (1978). The global optimisation problem: an introduction. In L.C.W. Dixon and G.P. Szego, editors, *Towards Global Optimisation 2*, 1–15. North Holland, New York.

J.F. Elder IV (1992). Global R^d optimization when probes are expensive: the GROPE algorithm. *Proceedings of the 1992 IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 1, 577–582, Chicago.

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

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.

J. Koehler and A. Owen. (1996). Computer experiments. In S. Ghosh and C.R. Rao, editors, *Handbook of Statistics, 13: Design and Analysis of Experiments*, 261–308. North Holland, New York.

H.J. Kushner (1964). A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86, 97–106.

M. Locatelli (1997). Bayesian algorithms for one-dimensional global optimization. *Journal of Global Optimization* 10, 57–76.

J. Mockus. Application of Bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization*, 4, 347–365, 1994.

C. Perttunen (1991). A computational geometric approach to feasible region division in constrained global optimization. *Proceedings of the 1991 IEEE Conference on Systems, Man, and Cybernetics*.

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

Sasena, M.J., Papalambros, P.Y. and Goovaerts, P. Metamodeling sampling criteria in a global optimzation framework. In *Proceedings of the* 8^{th} *AIAA / USAF / NASA / ISMMO Symposium of Multidisciplinary Analysis & Optimization*, September 2000. AIAA Paper 2000-4921.

Schonlau, M., Welch, W.J. and Jones, D.R. Global versus local search in constrained optimization of computer models. In N. Flournoy, W.F. Rosenberger, and W.K. Wong, editors, *New Developments and Applications in Experimental Design*, Institute of Mathematical Statistics. Also available as Technical Report RR-97-11, Institute for Improvement in Quality and Productivity, University of Waterloo, Waterloo, Ontario, CANADA, December 1997.

B.E. Stuckman (1988). A global search method for optimizing nonlinear systems. *IEEE Transactions on Systems, Man, and Cybernetics* 18, 965–977.

Theil, H. (1971). *Principles of Econometrics*. John Wiley, New York.

Torn, A. and Zilinskas, A. (1987). *Global Optimization*, Springer, Berlin.

A. Zilinskas (1992). A review of statistical models for global optimization. *Journal of Global Optimization* 2, 145–153.