New Developments and Applications in Experimental Design
IMS Lecture Notes – Monograph Series (1998) Volume 34

GLOBAL VERSUS LOCAL SEARCH IN CONSTRAINED OPTIMIZATION OF COMPUTER MODELS

BY MATTHIAS SCHONLAU, WILLIAM J. WELCH[1] AND DONALD R. JONES

University of Waterloo, University of Waterloo and General Motors

Engineering systems are now frequently optimized via computer models. The input-output relationships in these models are often highly nonlinear deterministic functions that are expensive to compute. Thus, when searching for the global optimum, it is desirable to minimize the number of function evaluations. Bayesian global optimization methods are well-suited to this task because they make use of all previous evaluations in selecting the next search point. A statistical model is fit to the sampled points which allows predictions to be made elsewhere, along with a measure of possible prediction error (uncertainty). The next point is chosen to maximize a criterion that balances searching where the predicted value of the function is good (local search) with searching where the uncertainty of prediction is large (global search). We extend this methodology in several ways. First, we introduce a parameter that controls the local-global balance. Secondly, we propose a method for dealing with nonlinear inequality constraints from additional response variables. Lastly, we adapt the sequential algorithm to proceed in stages rather than one point at a time. The extensions are illustrated using a shape optimization problem from the automotive industry.

1. Introduction. Global optimization via a computer model (sometimes called a computer code) is a problem encountered frequently in engineering. In this article, for example, we will discuss the optimization of the shape of an automobile piston. The inputs to the piston model are parameters describing the piston shape. The outputs are quality characteristics: undesirable piston motion (which causes noise) and the maximum pressure between the piston and the bore (which affects wear). The objective is to find the combination of shape parameters that minimizes maximum pressure subject to a constraint on motion. When function evaluations are fairly expensive, as here, there is a need to use optimization methods that require few evaluations. We shall see that the objective, maximum pressure, is highly nonlinear in the shape parameters; hence, some care is also necessary to find the global optimum.

---

[1] Research funded by the Natural Sciences and Engineering Research Council of Canada.
Received September 1997; revised December 1998.
AMS 1991 subject classifications. Primary 62L05, 65K10; Secondary 60G15, 62G07.
Key words and phrases. Bayesian global optimization, computer code, sequential design, stochastic process.

If the global optimum of a complex relationship is to be found with a limited number of computer-model runs, there has to be some modeling to predict behavior where the function has not been evaluated. Bayesian global optimization uses flexible statistical models, typically stochastic processes, which are capable of handling highly nonlinear relationships. After a few initial runs of the computer code, the statistical model for the objective function leads to a criterion for sequentially selecting new search points. For example, the algorithms of Kushner (1964), Perttunen and Stuckman (1992), and Zilinskas (1992) choose the next run to maximize the probability of improving the best evaluation so far. Mockus (1994) and Mockus, Tiesis, and Zilinskas (1978) used the expected improvement in the current best evaluation, a criterion that takes account of the magnitude of possible improvement. It balances the desire to search at locations with good predicted values (local search) with the desire to check where the uncertainty of prediction is large (global search).

Thus, in general terms, unconstrained Bayesian global optimization proceeds as follows:

1. Choose a small initial experimental design (set of points) spread over the entire input space. Run the computer code at these points.

2. Use all previous function evaluations to fit a statistical model for the objective function.

3. Based on the fitted model, find the "most promising" point in the input space for the next run.

4. Compute a stopping criterion. If it is met, then stop.

5. Run the computer code at the selected point in the input space. Go to Step 2.

This algorithm for unconstrained optimization is described in more detail in Schonlau (1997). He showed that, by making several adaptations to previous Bayesian algorithms, one could reliably optimize the functions in a well-known suite of test problems using fewer evaluations. The improved efficiency was due partly to the use of a Gaussian stochastic process model with a flexible correlation function, allowing these functions to be modeled more accurately. Moreover, by using diagnostic plots to guide the choice of a transformation of the response where necessary, the Gaussian stochastic process model was effective even for some fairly pathological problems in the test suite.

In Mockus, Tiesis, and Zilinskas (1978), the "most promising" point in Step 3 of the algorithm outlined above was the one with the largest expected improvement in the current minimum. Here, we extend the expected-improvement criterion in several ways. First, we generalize it to give more control over how global the search will be. Secondly, we propose a way of dealing with nonlinear constraints from additional response variables. Thirdly, we adapt the sequential algorithm so that runs can be made in stages of several points rather than one point at a time. This is particularly convenient if the optimization algorithm does not communicate directly with the computer model and manual intervention is required.

The outline of the paper is as follows. Section 2 describes the stochastic-process model. In Section 3 we derive a generalized expected-improvement criterion that allows control of how global the search is. We also describe a stopping rule. Section 4 explains how we deal with constraints. In Section 5 we adapt the methodology to allow sequential design in stages. Section 6 illustrates these ideas on the piston-shape problem. Finally, Section 7 concludes with some discussion.

2. Modeling approach. Suppose after an initial experimental design (set of sampled points) or at some later iteration of the algorithm, we have n vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ at which the response $y(\mathbf{x})$ has been evaluated. Each vector $\mathbf{x}$ is d-dimensional for the d inputs (explanatory variables) $x_1, \ldots, x_d$. The corresponding output values for a given response variable are denoted by $\mathbf{y} = (y_1, \ldots, y_n)^T$. Following the approach of, for example, Sacks, Welch, Mitchell, and Wynn (1989), the response is treated as a random function or a realization of a Gaussian stochastic process:

$$Y(\mathbf{x}) = \beta + Z(\mathbf{x}), \tag{2.1}$$

where $E[Z(\mathbf{x})] = 0$ and $\mathrm{Cov}[Z(\mathbf{x}), Z(\mathbf{x}')] = \sigma^2 R(\mathbf{x}, \mathbf{x}')$ for two input vectors $\mathbf{x}$ and $\mathbf{x}'$.

The correlation function $R(\cdot, \cdot)$ is crucial to this approach. Here it is assumed to have the form:

$$R(\mathbf{x}, \mathbf{x}') = \prod_{j=1}^{d} \exp(-\theta_j |x_j - x_j'|^{p_j}), \tag{2.2}$$

where $\theta_j \geq 0$ and $0 < p_j \leq 2$. In each coordinate direction, larger p_j can be interpreted as a parameter increasing the smoothness of the response surface, while larger θ_j indicates greater activity or nonlinearity.

This model leads to a best linear unbiased predictor and an associated mean squared error. For given correlation parameters $\theta_1, \ldots, \theta_d$ and $p_1, \ldots, p_d$ in (2.2), the predictor of y at an untried $\mathbf{x}$ can be shown to be:

$$\hat{y}(\mathbf{x}) = \hat{\beta} + \mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\beta}), \tag{2.3}$$

where $\mathbf{r}(\mathbf{x})$ is the $n \times 1$ vector of correlations $R(\mathbf{x}, \mathbf{x}_i)$ for $i = 1, \ldots, n$ between Z at $\mathbf{x}$ and at each of the n sampled points, $\mathbf{R}$ is the $n \times n$ matrix of correlations $R(\mathbf{x}_i, \mathbf{x}_{i'})$ for $i, i' = 1, \ldots, n$ between the Z's at the sampled points, $\hat{\beta} = (\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1})^{-1}\mathbf{1}^T\mathbf{y}$ is the generalized least squares estimator of β, and $\mathbf{1}$ is a vector of 1's. The mean squared error (MSE) of this predictor can be derived as:

$$\mathrm{MSE}[\hat{y}(\mathbf{x})] \equiv s^2(\mathbf{x}) = \sigma^2 \left[1 - (1 \quad \mathbf{r}^T(\mathbf{x})) \begin{pmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \mathbf{R} \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ \mathbf{r}(\mathbf{x}) \end{pmatrix}\right]. \tag{2.4}$$

The predictor interpolates the observed responses, and the MSE is zero at the design points $\mathbf{x}_1, \ldots, \mathbf{x}_n$.

In practice, the correlation parameters, $\theta_1, \ldots, \theta_d$ and $p_1, \ldots, p_d$ in (2.2), and σ^2 have to be estimated; we use maximum likelihood estimation. In this way, the correlation function $R(\cdot, \cdot)$ and hence the properties of the fitted predictor (smoothness,

etc.) are tuned to the data. The MSE in (2.4) ignores the variability from replacing these parameters by their estimates. Often, though, this extra source of uncertainty is relatively small and (2.4) provides a realistic estimate of prediction error [e.g., Welch, Buck, Sacks, Wynn, Mitchell, and Morris (1992)].

The predictor (2.3) based on the Gaussian stochastic process model with correlation function (2.2) has proven to be accurate for numerous applications; see, for example, Currin, Mitchell, Morris, and Ylvisaker (1991), Sacks, Schiller, and Welch (1989), Sacks, Welch, Mitchell, and Wynn (1989), and Welch et al. (1992). For optimization, the results presented by Schonlau (1997) show that this model is very competitive compared with other stochastic processes, for example Wiener processes, or with other correlation functions.

3. Generalized expected-improvement criterion. In this section we derive a criterion for determining the next sampling location. It generalizes the criterion proposed by Mockus, Tiesis, and Zilinskas (1978), which can be interpreted as the expected improvement in the minimum y value found so far when the next function evaluation is made.

We extend the expected-improvement criterion to include an additional integer-valued parameter, g. The larger the value of g, the more globally will the algorithm tend to search. Let $y_{\min}$ denote the minimum amongst the output values $y_1, \ldots, y_n$ so far. If the function is evaluated at $\mathbf{x}$ to give $y(\mathbf{x})$, then the improvement in $y_{\min}$ raised to the integer power $g \geq 0$ is

$$I^g(\mathbf{x}) = \begin{cases} [y_{\min} - y(\mathbf{x})]^g & \text{if } y(\mathbf{x}) < y_{\min} \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

Uncertainty about the unknown $y(\mathbf{x})$ is represented by saying it is has a normal distribution with mean given by the predictor $\hat{y}(\mathbf{x})$ in (2.3) and variance given by $s^2(\mathbf{x})$ in (2.4). Thus, $[y(\mathbf{x}) - \hat{y}(\mathbf{x})]/s(\mathbf{x})$ is standard normal. Hereafter, for notational simplicity, we will usually suppress the dependence of y, etc. on $\mathbf{x}$.

For $g = 0$ taking the expectation yields the probability of improvement:

$$\mathrm{E}(I^0) = P(y < y_{\min}) = P(\frac{y - \hat{y}}{s} < u) = \Phi(u),$$

where

$$u = (y_{\min} - \hat{y})/s \tag{3.2}$$

is the normalized $y_{\min}$, and $\Phi()$ denotes the cumulative distribution function of the standard normal distribution.

For $g = 1, 2, \ldots$ it is shown in the Appendix that the generalized expected improvement is

$$\mathrm{E}(I^g) = s^g \sum_{k=0}^{g} (-1)^k \binom{g}{k} u^{g-k} T_k, \tag{3.3}$$

where

$$T_0 = \Phi(u) \quad \text{and} \quad T_1 = -\phi(u),$$

and T_k for $k > 1$ can be computed recursively from

$$T_k = -u^{k-1}\phi(u) + (k-1)T_{k-2}. \tag{3.4}$$

Although it is possible to write down a (fairly complicated) explicit expression for T_k, the above recurrence relationship is simpler to implement. Strictly speaking, $\mathrm{E}(I^g)$ is the *estimated* generalized expected improvement, as the predictor and its MSE are estimates.

The factor s^g in (3.3) gives the root mean squared error of prediction, s, greater weight as g increases. Because s is zero at sampled points and is largest in regions remote from all sampled points, increasing g tends to make the search more global. Another way of looking at this is that in general there is a tradeoff in choosing between small improvement with large probability (local search) versus large improvement with small probability (global search). As g increases larger improvements become more important, even if they have small probability, and the search is more global.

For the special cases $g = 1, 2, 3$ and $s > 0$ we obtain from (3.3):

$$\begin{aligned} \mathrm{E}(I) &= s[u\Phi(u) + \phi(u)] \\ \mathrm{E}(I^2) &= s^2[(u^2+1)\Phi(u) + u\phi(u)] \\ \mathrm{E}(I^3) &= s^3[(u^3+3u)\Phi(u) + (u^2+2)\phi(u)]. \end{aligned} \tag{3.5}$$

The choice of $g = 1$ reproduces the expected-improvement criterion used by Schonlau (1997). The case $g = 2$ is interesting as

$$\mathrm{E}(I^2) = [\mathrm{E}(I)]^2 + \mathrm{Var}(I).$$

Thus, $g = 2$ gives a monotonic transformation of the original expected-improvement criterion, $\mathrm{E}(I)$, plus the variability in the improvement, $\mathrm{Var}(I)$. It explicitly takes account of the uncertainty of improvement (up to estimation of the correlation parameters).

When $g = 1$, we base the stopping criterion on $\mathrm{E}(I)$. If the maximum $\mathrm{E}(I)$ over $\mathbf{x}$ is smaller than a prespecified tolerance, we stop. For $g > 1$, we compare the maximum $[\mathrm{E}(I^g)]^{1/g}$ with the tolerance. Since I is nonnegative and I^g is a convex function of I for $I \geq 0$, Jensen's inequality applies and yields $[\mathrm{E}(I^g)]^{1/g} > \mathrm{E}(I)$. Assuming the same tolerances, stopping rules based on $[\mathrm{E}(I^g)]^{1/g}$ will tend to sample more points and be more conservative.

The parameter g is a systematic way of controlling the global versus local trade-off. Hopefully, this avoids ad hoc approaches, as used for example by Mockus, Tiesis, and Zilinskas (1978), where the recommendation was to inflate the MSE by some factor to avoid too local a search. (The necessity to adjust the MSE may have resulted from these authors' use of a less flexible correlation function, rather than from the expected-improvement criterion itself.)

The generalized expected improvement, $\mathrm{E}(I^g)$, in (3.3) is a function of $\mathbf{x}$ that must be optimized to choose the location of the next run. We have tried several algorithms. DIRECT [Jones, Perttunen, and Stuckman (1993)] is an optimizer that will find the

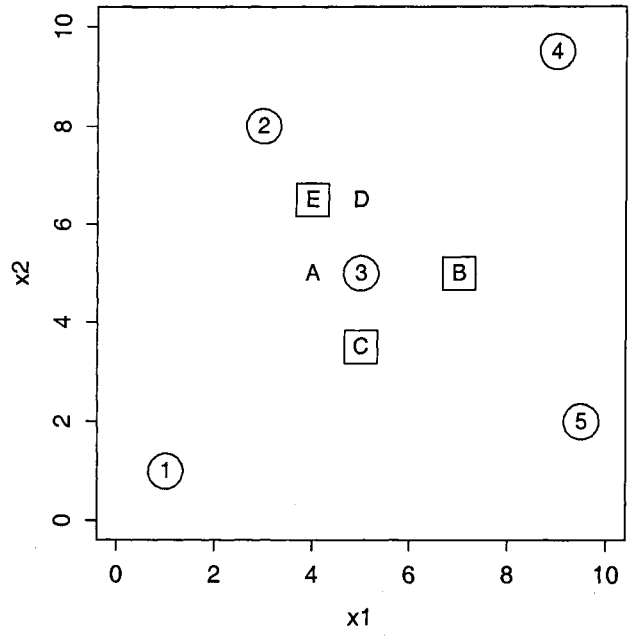

FIG. 1. *Starting points (B, C, and E) for the simplex search around design point 3. Points 1, 2, 4, and 5 are also design points, while points A and D are candidate starting points that are not chosen.*

global optimum given a fairly large number of function evaluations. This demonstrates the advantage of replacing the computationally expensive computer model with a statistical model: $\mathrm{E}(I^g)$ is relatively very easy to compute, and it is possible to carry out an extensive search.

We have also used multiple starts of the simplex algorithm [Nelder and Mead (1965)] to maximize $\mathrm{E}(I^g)$. At any existing design point $\mathbf{x}_i$, the root mean squared error s in (2.4) is zero, and hence so is $\mathrm{E}(I^g)$ in (3.3). Thus, good starting points for maximizing $\mathrm{E}(I^g)$ over $\mathbf{x}$ are locations between the existing design points. When searching for $\mathbf{x}_{n+1}$, we make n searches with search i started between $\mathbf{x}_i$ and its neighbors.

The simplex method requires $d+1$ starting points (recall that $\mathbf{x}$ has d dimensions) for search i. Starting point j for $j = 1, \ldots, d$ is $\mathbf{x}_i$ with only coordinate j changed. The remaining point has all coordinates perturbed. Figure 1 illustrates for $d = 2$. There are five design points labelled 1,..., 5 in circles. Consider the starting points for the search around design point 3. For the first starting point, only the x_1 coordinate is changed. The design points labelled 2 and 4 are the closest to 3 in this dimension, to the left and right, respectively. The point labelled A has an x_1 value halfway between points 2 and 3; similarly point B is halfway between points 3 and 4. The remaining coordinates of design point 3 are unchanged in generating points A and B. One of A or B is chosen at random as a starting point: say B, marked with a square. Similarly,

in the x_2 coordinate we randomly choose between the points C and D, these being halfway towards the closest design points, 5 and 2, respectively. Again, one of these is randomly chosen, say C. The final starting point is E. It takes the x_1 coordinate from A, and the x_2 coordinate from D, i.e., it uses the directions not chosen previously. The three points B, C, and E would be used to start the simplex search.

A similar procedure is used to start the searches around the remaining design points. When generating the starting points from design point 4 in Figure 1, however, there is no design point with a larger x_2 value. Here we would generate a candidate starting point halfway to the x_2 boundary.

In this way, the $d+1$ starting points infiltrate the spaces around $\mathbf{x}_i$. This is the method used for the example in Section 6.

Work on an exact branch and bound algorithm to maximize $\mathrm{E}(I^g)$ is underway and will be reported elsewhere. These three methods—DIRECT, simplex with multiple starts, and branch and bound—appear to give similar results in terms of total number of function evaluations, etc.

4. Minimization subject to constraints. In this section we consider the problem of minimizing an objective function subject to constraints on additional response variables. A strategy is offered treating the predictions for all response variables as statistically independent. The strategy for the dependent case is outlined but would require the specification of the correlation structure between responses.

Denote the k response functions acting as constraints by $c_1(\mathbf{x}), \ldots, c_k(\mathbf{x})$ and suppose we want to minimize $y(\mathbf{x})$ subject to $a_i < c_i(\mathbf{x}) < b_i$ for $i = 1, \ldots, k$. We define the generalized improvement subject to constraints as

$$I_c^g(\mathbf{x}) = \begin{cases} [y_{\min} - y(\mathbf{x})]^g & \text{if } y(\mathbf{x}) < y_{\min} \text{ and } a_i \le c_i(\mathbf{x}) \le b_i \text{ for } i = 1, \ldots, k \\ 0 & \text{otherwise,} \end{cases}$$

where $y_{\min}$ is the minimum *feasible* value of the objective, y, amongst the current n runs. Again, we will suppress dependence on $\mathbf{x}$. Taking the expectation, we have

$$\mathrm{E}(I_c^g) = \int_{a_k}^{b_k} \cdots \int_{a_1}^{b_1} \int_{-\infty}^{y_{\min}} (y_{\min} - y)^g h(y, c_1, \ldots, c_k)\, dy\, dc_1 \cdots dc_k, \tag{4.1}$$

where $h(y, c_1, \ldots, c_k)$ is the joint density function of the $k+1$ response variables. This integral could be evaluated numerically if $h(\cdot)$ were available. Again, we represent uncertainty about y and $c_1, \ldots, c_k$ by taking $h(\cdot)$ to be (multivariate) normal. The means are given by the predictor $\hat{y}$ in (2.3) and the variances by s^2 in (2.4). There will be a maximum likelihood fit, predictor, and MSE for each response variable; other than the extra computational burden, this is tractable. The covariances between predictions for pairs of responses are not so tractable, however. It may be possible to borrow some ideas from the cokriging literature [e.g., Cressie (1993, Chapter 3.2.3)], but this would introduce further parameters to be estimated and make the computations very demanding.

We circumvent this practical issue by treating the response variables as statistically independent. The constrained generalized expected improvement (4.1) then simplifies

to

$$\mathrm{E}(I_c^g) = \mathrm{E}(I^g)P(a_1 < c_1 < b_1)\cdots P(a_k < c_k < b_k). \tag{4.2}$$

That is, the generalized expected improvement in (3.3) is multiplied by the probabilities that each constraint is met. These probabilities are simply computed from the standard normal cumulative distribution function assuming each c_j has mean given by the predictor (2.3) and variance given by the MSE in (2.4).

5. Sequential design in stages. Unless the computer model that generates function evaluations is directly linked to the expected-improvement optimization algorithm, sampling one point at a time is inconvenient. Here, we consider a multi-stage algorithm, sampling q points at a time.

After n points, if we sample q further points, $\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+q}$, to give response values $y_{n+1}, \ldots, y_{n+q}$, the generalized improvement (3.1) becomes

$$I^g = [\max(0, y_{\min} - y_{n+1}, \ldots, y_{\min} - y_{n+q})]^g.$$

Taking the expectation of this quantity would require numerical evaluation of a multivariate normal probability, a computationally demanding task that would have to be repeated many times when searching for the optimal $\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+q}$.

In light of this, we again resort to some simplifications for practical computation. First, we optimize the q points sequentially one at a time (even though all q runs of the computer code will be made together). Thus, once $\mathbf{x}_{n+1}$ is optimized it is fixed when optimizing $\mathbf{x}_{n+2}$, etc. For point $\mathbf{x}_{n+i}$, the generalized expected improvement in (3.3) becomes

$$E_{n+i}(I^g) = s_{n+i-1}^g \sum_{k=0}^{g} (-1)^k \binom{g}{k} u^{g-k} T_k, \tag{5.1}$$

where T_k is defined as in (3.4). The root mean squared error of prediction, s, in (2.4) depends only on the design points, not on the response values. (The responses are used implicitly in estimation of the stochastic process parameters, but they are only re-estimated when all q runs of the computer model are made.) Thus, when optimizing $\mathbf{x}_{n+i}$, we can use s_{n+i-1}, the root mean squared error from $\mathbf{x}_1, \ldots, \mathbf{x}_{n+i-1}$. As s_{n+i-1} and hence $E_{n+i}(I^g)$ in (5.1) are small near any of the design points $\mathbf{x}_1, \ldots, \mathbf{x}_{n+i-1}$, the new point will avoid previously sampled locations. Some trial and error with the algorithm, however, showed that u in (3.2) and hence T_k in (3.4) should be based on only the first n points, i.e., $u = (y_{\min} - \hat{y})/s_n$. The predictor $\hat{y}$ in (2.3) cannot be updated until the new runs are actually made, and s_n is the appropriate normalizing root mean squared error.

Similarly, if constraint functions are present, the feasibility probabilities in (4.2) are computed using s_n.

6. Example: Engineering design of a piston. To illustrate these methods we take an example from automotive engineering involving the optimization of the shape of a piston. There are two output variables: maximum pressure between the piston and the bore ($p_{\max}$, measured in MPa) and undesirable piston motion (m_p, measured

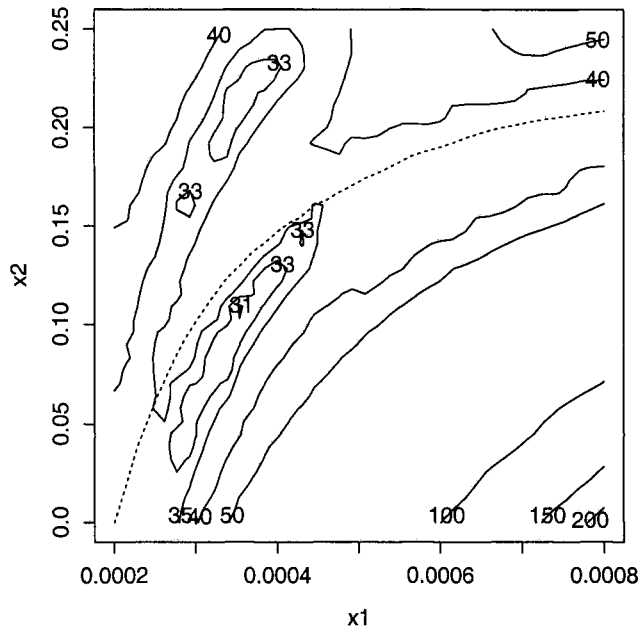

FIG. 2. *Contours of the true objective function, $p_{\max}$ (MPa). The dotted contour line denotes the boundary of the constraint, $m_p < 30\mu$m.*

in μm). These are related to engine wear and noise, respectively. We want to minimize $p_{\max}$ subject to $m_p < 30\mu$m. The objective and constraint functions are related to two inputs (x_1 and x_2) that describe the shape of the piston.

The computer model has further input variables and responses, but here we have limited the problem. By keeping all but the two most important explanatory variables fixed we can easily visualize the sequential design strategy. Moreover, for the simplified problem it is possible to sample the input space very densely, determine the true optimum, and hence illustrate that the algorithm does indeed find the global minimum with a more limited number of runs.

Figure 2 gives a contour plot of the true objective function, $p_{\max}$, from running the computer code on a 20×20 grid of x_1 and x_2 values. It shows that $p_{\max}$ is very nonlinear with numerous local optima. Similarly, Figure 3 depicts the true constraint function, m_p. The boundary of the constraint $m_p < 30\mu$m is also shown in Figure 2. The global unconstrained minimum for $p_{\max}$ is 30.117 MPa, located at $(0.00035, 0.103)$, but this is just outside the $m_p < 30\mu$m constraint boundary. The lowest value in the feasible region is $p_{\max} = 31.323$ MPa, at $(0.00035, 0.214)$.

Initially, we evaluate the function at 21 sites based on a Latin hypercube experimental design [McKay, Conover, and Beckman (1979)]. Typically, we use initial designs with the number of runs equal to 10 times the number of variables. With 21 runs, the

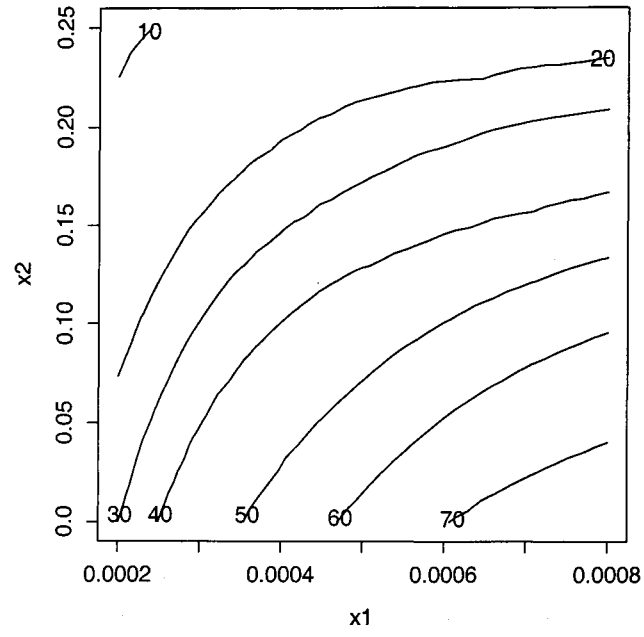

FIG. 3. *Contours of the true constraint function, $m_p(\mu m)$.*

spacing in each coordinate is conveniently 5% of the range.

Schonlau (1997) proposed the use of diagnostic plots after the initial experimental design to assess whether a function is modeled well by the Gaussian stochastic process (2.1). If not, a transformation may help. The diagnostic plots (not shown here), which are all based on cross validation, indicate that m_p is modeled with very good accuracy. The response $p_{\max}$ is much more difficult to model, however. This is apparent from the predictor in Figure 4, which suggests a highly nonlinear, multimodal function. Compared with the true function in Figure 2, the predictor shows a fair amount of discrepancy. In practice, we would not know the true function, and we would have to cross validate the predictor using only the sampled values. The prediction errors are again fairly large, but they are commensurate with the root mean squared errors from (2.4). The other diagnostic plots are also satisfactory. Thus, it seems that the large prediction error is due to sparse sampling of a very complex function, and the model (2.1) captures these difficulties.

Because the predictor for $p_{\max}$ in Figure 4 indicates a complex objective function, we proceed cautiously with the optimization. We take $g = 2$ in the generalized expected-improvement criterion, so that the search is fairly global, and compute $\mathrm{E}(I^2)$ from (3.5). To deal with the constraint, we multiply $\mathrm{E}(I^2)$ by the probability that $m_p < 30\mu$m, as in (4.2).

First, we optimize one point at a time; Figure 5 shows the initial 21-run design and

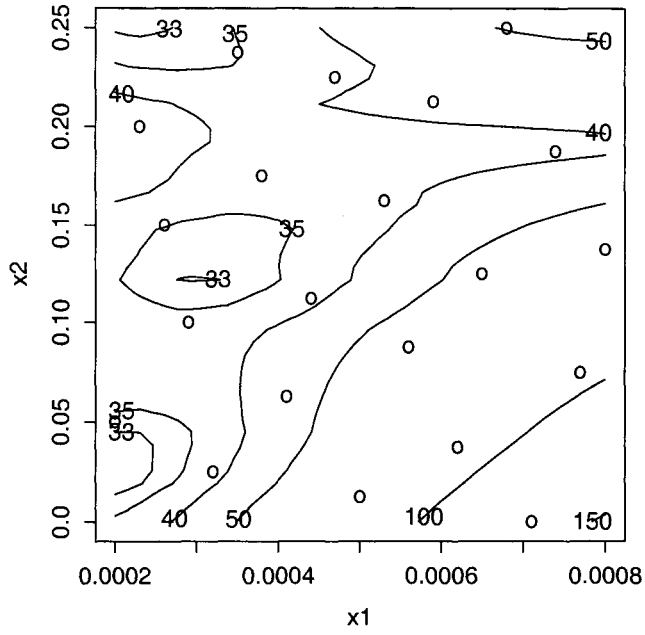

FIG. 4. *Predictor for $p_{\max}$ (MPa) after the initial 21-run experimental design (shown by circles).*

the points sequentially introduced in the search. In total, 89 runs are made. The figure distinguishes runs 1–21 (the initial design), 22–40, 41–60, and 61–89, so that we may see how the algorithm chooses search points as it progresses. We have also included contours of the true objective function to highlight the local minima, and the constraint boundary to show the feasible region. The first points chosen by the algorithm (runs 22–40, shown as squares) are spread widely in the feasible region, in a global search. There is some concentration at the lower left corner close to the boundary of the m_p constraint (shown by a dotted contour), where the objective function is fairly low. Runs 41–60 (triangles) locate two local minima and sample them intensively. Finally, runs 61–89 (pluses) focus mainly on one of these local minima, which is the constrained global minimum. Virtually no points are sampled outside of the feasibility region because the constraint function is modeled very well.

After 89 points, a stopping criterion with a relative tolerance of .0001 is met. At termination, the actual relative tolerance is smaller than 10^{-7}.

To illustrate the sequential design in stages we also try four stages of 10 points each (after the initial 21 runs). Four stages would be reasonably convenient in practice. The expected improvement is computed using (5.1) with again $g = 2$.

Table 1 gives the minimum feasible $p_{\max}$ value found after the initial 21 points and after each of the four stages. For comparison, the minima after the same numbers of runs when searching one point at a time are also given. Perhaps surprisingly, the

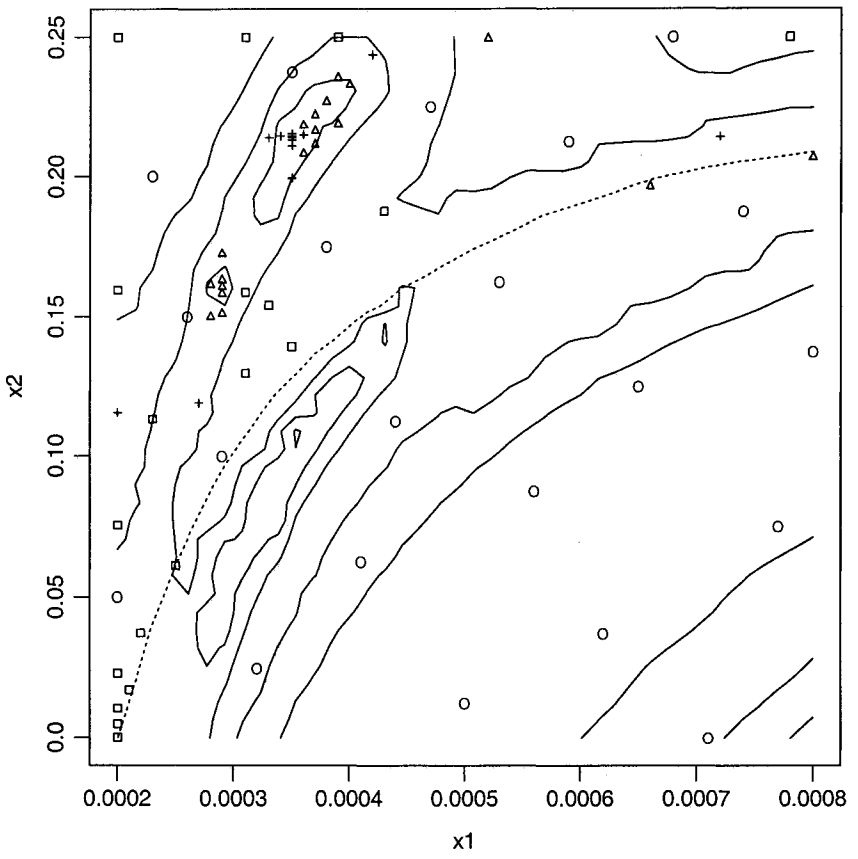

FIG. 5. *Initial experimental design (circles) and points introduced by the sequential minimization algorithm (squares for runs 22–40; triangles for runs 41–60; and pluses for runs 61–89). The contours shown are the true objective function, $p_{\max}$ (MPa). The dotted contour line denotes the boundary of the constraint, $m_p < 30\mu m$.*

TABLE 1

Minimum feasible $p_{\max}$ value found after 21, 31, ..., 61 function evaluations when searching one point at a time compared with minimization in stages of 10 runs.

		Minimum $p_{\max}$ (MPa)	
Stage	n	One point at a time	10 points at a time
Initial	21	34.06	34.06
1	31	33.70	32.15
2	41	32.42	32.00
3	51	31.94	32.00
4	61	31.94	31.76

minimization in stages does better after Stages 1, 2, and 4. The second stage (runs 32–41) includes several points close to the constrained global minimum, whereas searching one point at a time does not place a point in the vicinity of the constrained global minimum until about run 50. Inspection of the plot analogous to Figure 5 shows that the search in stages is even more global. With a highly nonlinear, multimodal objective a more global search is probably advantageous early on. In any case, there seems to be little lost by designing in stages in this example.

7. Discussion. For unconstrained global optimization, the expected-improvement algorithm appears to be very competitive in terms of function evaluations. Schonlau (1997) presented favorable comparisons with other methods for test problems with up to six input variables. The stochastic-process model underlying the algorithm has been used with higher-dimensional input. For example, Aslett, Buck, Duvall, Sacks, and Welch (1998) described a circuit-simulation problem with 36 input variables. Several competing responses (quality characteristics) led to a constrained optimization. In this example, the fitted models were used in a fairly ad hoc way to guide a sequence of experiments to optimize the circuit. Our ultimate goal is to extend the expected-improvement algorithm to deal with engineering problems of this magnitude in a more automatic way.

In this article we have taken several steps towards this goal. An analysis of an initial experiment might show that the optimization problem is likely to be difficult, because the objective function seems multimodal for instance. In such cases we would want to proceed cautiously in an automatic search. The extension of the expected-improvement criterion to control the local-global balance allows a more cautious global search. We have also outlined methods for dealing with constraints from multiple output variables and for running the computer model in stages.

Some difficulties were side-stepped along the way. Further work is necessary to model the relationships between response variables, where major trade-offs are often present. Similarly, the criterion for optimization in stages was simplified for computational reasons. Other issues, not addressed here, include dealing with variability from input factors representing manufacturing noise, etc. There is ongoing work on optimization of the expected-improvement criterion, to deal with computer models with many input variables.

Appendix. In this Appendix we derive the equations (3.3) and (3.4). For $s > 0$, we can rewrite the improvement given in (3.1) as

$$I^g = \begin{cases} s^g(u-v)^g & \text{if } v < u \\ 0 & \text{otherwise,} \end{cases}$$

where $u = (y_{\min} - \hat{y})/s$ and $v = (y - \hat{y})/s$.

Taking the expectation yields

$$\mathrm{E}(I^g) = s^g \int_{-\infty}^{u} \sum_{k=0}^{g} (-1)^k \binom{g}{k} u^{g-k} v^k \phi(v)\, dv = s^g \sum_{k=0}^{g} (-1)^k \binom{g}{k} u^{g-k} T_k,$$

where

$$T_k = \int_{-\infty}^{u} v^k \phi(v)\, dv,$$

and $\phi()$ is the standard normal density. This follows because v is standard normal.

We now calculate T_k using the partial integration technique, splitting the integrand up into v^{k-1} and $v\phi(v) = -\phi'(v)$:

$$T_k = -\left[v^{k-1}\phi(v)\right]_{-\infty}^{u} + (k-1)\int_{-\infty}^{u} v^{k-2}\phi(v)\, dv = -u^{k-1}\phi(u) + (k-1)T_{k-2}.$$

This establishes (3.3) and the recursive formula for T_k in (3.4). Since T_k is a function of T_{k-2}, two starting values are needed:

$$T_0 = \int_{-\infty}^{u} \phi(v)dv = \Phi(u) \quad \text{and} \quad T_1 = \int_{-\infty}^{u} v\phi(v)\, dv = -\left[\frac{\exp(-v^2/2)}{\sqrt{2\pi}}\right]_{-\infty}^{u} = -\phi(u),$$

where $\Phi()$ is the standard normal cumulative distribution function.

Acknowledgments. We thank Lara Wolfson and two referees for comments.

REFERENCES

ASLETT, R., BUCK, R.J., DUVALL, S.G., SACKS, J., AND WELCH, W.J. (1998). Circuit optimization via sequential computer experiments: design of an output buffer. *Appl. Statist.* **47** 31–48.

CRESSIE, N.A.C (1993). *Statistics for Spatial Data.* Wiley, New York.

CURRIN, C., MITCHELL, T., MORRIS, M., AND YLVISAKER, D. (1991). Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *J. Amer. Statist. Assoc.* **86** 953–963.

JONES, D.R., PERTTUNEN, C.D., AND STUCKMAN, B.E. (1993). Lipschitzian optimization without the Lipschitz constant. *J. Optim. Theor. Appl.* **79** 157–181.

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

MCKAY, M.D., CONOVER, W.J., AND BECKMAN, R.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. In *Towards Global Optimisation 2* (L.C.W. Dixon and G.P. Szego, eds.) 117-129. North Holland, Amsterdam.

MOCKUS, J. (1994). Application of Bayesian approach to numerical methods of global and stochastic optimization. *J. Global Optim.* **4** 347–365.

NELDER, J.A. AND MEAD, R. (1965). A simplex method for function minimization. *Comput. J.* **7** 308–313.

PERTTUNEN, C.D. AND STUCKMAN, B.E. (1992). The normal score transformation applied to a multi-univariate method of global optimization. *J. Global Optim.* **2** 167–176.

SACKS, J., SCHILLER, S.B., AND WELCH, W.J. (1989). Designs for computer experiments. *Technometrics* **31** 41–47.

SACKS, J., WELCH, W.J., MITCHELL, T.J., AND WYNN, H.P. (1989). Design and analysis of computer experiments (with discussion). *Statist. Sci.* **4** 409–435.

SCHONLAU, M. (1997). Computer experiments and global optimization. University of Waterloo, Ontario (doctoral thesis).

WELCH, W.J., BUCK, R.J., SACKS, J., WYNN, H.P., MITCHELL, T.J., AND MORRIS, M.D. (1992). Screening, predicting, and computer experiments. *Technometrics* **34** 15–25.

ZILINSKAS, A. (1992). A review of statistical models for global optimization. *J. Global Optim.* **2** 145–153.

MATTHIAS SCHONLAU
NATIONAL INSTITUTE
FOR STATISTICAL SCIENCES
P.O. BOX 14006
RESEARCH TRIANGLE PARK
NORTH CAROLINA 27709-4006 U.S.A.

WILLIAM J. WELCH
DEPARTMENT OF STATISTICS AND ACTUARIAL SCIENCE
INSTITUTE FOR IMPROVEMENT
IN QUALITY AND PRODUCTIVITY
UNIVERSITY OF WATERLOO
WATERLOO, ONTARIO N2L 3G1
CANADA

DONALD R. JONES
GENERAL MOTORS R&D CENTER
MAIL CODE 480-106-359
30500 MOUND ROAD
WARREN, MICHIGAN 48090-9055 U.S.A.